SGGS Decision Procedures

Maria Paola Bonacina and Sarah Winkler(✉)

Università degli Studi di Verona, Verona, Italy
{mariapaola.bonacina,sarahmaria.winkler}@univr.it

Abstract. SGGS (Semantically-Guided Goal-Sensitive reasoning) is a conflict-driven first-order theorem-proving method which is refutationally complete and model complete in the limit. These features make it attractive as a basis for decision procedures. In this paper we show that SGGS decides the *stratified fragment* which generalizes EPR, the *PVD fragment*, and a new fragment that we dub *restrained*. The new class has the *small model property*, as the size of SGGS-generated models can be upper-bounded, and is also decided by hyperresolution and ordered resolution. We report on experiments with a termination tool implementing a restrainedness test, and with an SGGS prototype named Koala.

1 Introduction

Many applications of automated reasoning require to combine the decidability of satisfiability with an expressive logic. Since first-order theorem proving is only semidecidable, the quest for decidable fragments of first-order logic is key in advancing the field, and many classes of formulæ were shown decidable. Without claiming completeness (see [15,18,22] for surveys), we mention: the *Bernays-Schönfinkel class*, also known as EPR for *effectively propositional* [3,8,19,37,39]; the *Ackermann class* [2,23]; the *monadic class* with and without equality [2,5,23]; the *positive variable dominated* (PVD) fragment [17]; the *two-variable* fragment (FO^2) [21]; the *guarded* fragment [4,20]; the *modal fragment* [9,31], which is included in the EPR [22], FO^2 [32], and guarded [4] fragments; and the *stratified* fragment [1,25,36], which generalizes EPR. However, many theorem proving problems from the practice fall in none of these classes.

Example 1. Problem HWV036-2 from TPTP 7.3.0 [41] specifies a full-adder in 51 clauses, including for instance:

$$\neg\mathsf{and}_{\mathsf{ok}}(x) \vee \neg\mathsf{1}(\mathsf{in}_1(x)) \vee \neg\mathsf{1}(\mathsf{in}_2(x)) \vee \mathsf{1}(\mathsf{out}_1(x)), \quad \neg\mathsf{lor}(x) \vee \mathsf{or}_{\mathsf{ok}}(x) \vee \mathsf{error}(x),$$
$$\neg\mathsf{halfadd}(x) \vee \mathsf{connection}(\mathsf{in}_1(x), \mathsf{in}_1(\mathsf{or}_1(x))), \quad \neg\mathsf{fulladd}(x) \vee \mathsf{halfadd}(\mathsf{h}_1(x)).$$

This set is satisfiable, which means that termination of a theorem prover is *a priori* not guaranteed. However, it is neither EPR ($\exists^*\forall^*\varphi$ formulæ with φ quantifier- and function-free), nor Ackermann ($\exists^*\forall\exists^*\varphi$ formulæ with φ as above) nor FO^2 (only two variables, no functions), nor monadic (only unary predicates, no functions). One can also check that it is neither guarded nor stratified.

This research was funded in part by grant "Ricerca di base 2017" of the Università degli Studi di Verona. Authors are listed alphabetically.

N. Peltier and V. Sofronie-Stokkermans (Eds.): IJCAR 2020, LNAI 12166, pp. 356–374, 2020.
https://doi.org/10.1007/978-3-030-51074-9_20

As refutational completeness guarantees termination on unsatisfiable inputs, if one can prove termination of an inference system on satisfiable inputs in a certain class, any strategy given by that inference system and a fair search plan is a decision procedure for satisfiability in that class. Here we consider *Semantically-Guided Goal-Sensitive reasoning* (SGGS) [13,14], that is a refutationally complete instance-based theorem-proving method especially suitable for decision procedures: SGGS is *model-based* (it searches for a model by building candidates), *semantically guided* (the search is guided by a fixed initial interpretation), *conflict-driven* (it applies inferences such as resolution only to explain conflicts), *proof confluent* (it never needs to undo inferences), and *model complete in the limit* (if the input is satisfiable, the limit of the derivation represents a model), so that model generation is guaranteed if termination is.

This paper shows that SGGS decides the *stratified fragment* [1], which includes EPR and finds application in verification [1,25,36], and the *PVD fragment*. Then, we discover a *new decidable class* named the *restrained fragment*, and show that SGGS, ordered resolution, and hyperresolution all decide it. Since it is possible to compute *bounds on the size of SGGS-generated models*, this new class enjoys the *small model property*. We give a sufficient condition for membership in the restrained fragment that can be tested *automatically* by termination tools for rewriting. The relevance of this new class is evaluated empirically by applying this test to problems in TPTP. For instance, the axiomatization in Example 1, as well as all the TPTP problems including it, turn out to be restrained. We also summarize the outcomes of experiments with an SGGS prototype, named Koala, built reusing code from Konstantin Korovin's iProver [24,26].

The paper is structured as follows. Since the stratified fragment has sorts, Sect. 2 presents SGGS for a language with sorts. Section 3 shows that SGGS decides the stratified fragment. In Sect. 4 we define the restrained fragment, establish the small model property, and prove that SGGS decides both this class and PVD. Ordered resolution and hyperresolution also decide restrained sets (Sect. 5). The experimental results are reported in Sect. 6, and Sect. 7 concludes the paper.

2 Preliminaries: SGGS for Many-Sorted Logic

Let S be a set of clauses in many-sorted logic with non-empty sorts (there is a ground term for every sort). We use a, b for constants, P, Q for predicates, f, g for functions, w, x, y, z for variables, t, u for terms, L, M for literals, $at(L)$ for L's atom, C, D for clauses, $\mathcal{V}ar(C)$ for the set of variables in C, α, σ for substitutions, I, J for interpretations, and we extend the at notation to sets of literals, clauses, and sets of clauses. C^+ and C^- are the disjunctions of the positive and negative literals in C, respectively; C is *positive* if $C = C^+$ and *negative* if $C = C^-$.

In SGGS, a clause C may have a constraint A, written $A \rhd C$. An atomic *SGGS constraint* is *true*, *false*, $t \equiv u$, and $top(t) = f$, where $\equiv$ is syntactic identity, and $top(t)$ is the top symbol of term t. The negation, conjunction, and disjunction of constraints is a constraint. Constraints in *standard form* are *true*, *false*, and conjunctions of distinct atomic constraints $x \not\equiv y$ and $top(x) \neq f$. Substitutions are *sort-preserving* ($x\sigma$ has the same sort as x) so that instantiation respects sorts. The set $Gr(A \rhd C)$ of *constrained ground instances* (cgi) of $A \rhd C$ is the set of ground instances of C that satisfy A. Literals $A \rhd L$ and $B \rhd M$ *intersect* if $at(Gr(A \rhd L)) \cap at(Gr(B \rhd M)) \neq \emptyset$, and are *disjoint* otherwise.

Example 2. In a signature with sorts $\{s_1, s_2\}$ and symbols $\mathsf{a}\colon s_1$, $\mathsf{b}\colon s_2$, $\mathsf{f}\colon s_1 \rightarrow s_2$, and $\mathsf{P} \subseteq s_2 \times s_2$, the only term of sort s_1 is a, and b and $\mathsf{f}(\mathsf{a})$ are the only terms of sort s_2. Thus, $Gr(\mathsf{P}(x, y)) = \{\mathsf{P}(\mathsf{b}, \mathsf{b}), \mathsf{P}(\mathsf{f}(\mathsf{a}), \mathsf{b}), \mathsf{P}(\mathsf{b}, \mathsf{f}(\mathsf{a})), \mathsf{P}(\mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{a}))\}$. For $\mathsf{P}(\mathsf{f}(x), y)$, with $x\colon s_1$ and $y\colon s_2$, constraint $top(x) \neq \mathsf{a}$ is unsatisfiable, while $top(y) \neq \mathsf{a}$ is valid. Then, $top(x) \neq \mathsf{a} \rhd \mathsf{P}(\mathsf{f}(x), y)$ is equivalent to $false \rhd \mathsf{P}(\mathsf{f}(x), y)$ and has no cgi's, while $top(y) \neq \mathsf{a} \rhd \mathsf{P}(\mathsf{f}(x), y)$ is equivalent to $true \rhd \mathsf{P}(\mathsf{f}(x), y)$, or simply $P(\mathsf{f}(x), y)$, and has all cgi's, namely $P(\mathsf{f}(\mathsf{a}), \mathsf{b})$ and $P(\mathsf{f}(\mathsf{a}), \mathsf{f}(\mathsf{a}))$.

SGGS is *semantically guided* by an *initial interpretation* I: unless $I \models S$, SGGS seeks a model of S, by building candidate partial interpretations different from I, and using I as default to complete them. If the empty clause $\bot$ arises in the process, unsatisfiability is reported. If I is the *all-negative interpretation* I^- that makes all negative literals true, SGGS tries to discover which positive literals need to be true to satisfy S, and dually if I is the *all-positive interpretation* I^+. While I can be any Herbrand interpretation, I^+ and I^- suffice in this paper.

SGGS works with a *trail* of clauses $\Gamma = A_1 \rhd C_1[L_1], \ldots, A_n \rhd C_n[L_n]$, where $C[L]$ means that literal $L \in C$ is *selected* in C. The length of Γ and its prefix of length j are denoted $|\Gamma|$ and $\Gamma|_j$, respectively. An SGGS-trail Γ represents a partial interpretation $I^p(\Gamma)$: if Γ is empty, denoted ε, $I^p(\Gamma) = \emptyset$; otherwise, $I^p(\Gamma) = I^p(\Gamma|_{n-1}) \cup pcgi(A_n \rhd L_n, \Gamma)$, where pcgi abbreviates *proper constrained ground instances*. A pcgi of $A_n \rhd C_n[L_n]$ is a cgi $C[L]$ that is not satisfied by $I^p(\Gamma|_{n-1})$ (i.e., $I^p(\Gamma|_{n-1}) \cap C[L] = \emptyset$) and can be satisfied by adding L as $\neg L \notin I^p(\Gamma|_{n-1})$. For the selected literal, $pcgi(A_n \rhd L_n, \Gamma) = \{L \,:\, C[L] \in pcgi(A_n \rhd C_n[L_n], \Gamma)\}$. $I^p(\Gamma)$ is completed into an interpretation $I[\Gamma]$ by consulting I for the truth value of any literal undefined in $I^p(\Gamma)$.

A literal L is *uniformly false* in an interpretation J, if all $L' \in Gr(L)$ are false in J. Then, L is said to be I*-false* if it is uniformly false in I, and I*-true* if it is true in I. SGGS requires that if a clause in Γ has I-false literals, one is selected, so as to differentiate $I[\Gamma]$ from I. A clause whose literals are all I-true is an I*-all-true clause*, and only in such a clause an I-true literal is selected.

A *conflict clause* is one whose literals are all uniformly false in $I[\Gamma]$. SGGS ensures that every I-all-true clause $C[L]$ in Γ is either a conflict clause or the *justification* of its selected literal L, meaning that all literals of $C[L]$ except L are uniformly false in $I[\Gamma]$, so that L must be true in $I[\Gamma]$ to satisfy $C[L]$. In the latter case $C[L]$ is in the *disjoint prefix* of Γ, denoted $dp(\Gamma)$, which is the longest prefix such that $pcgi(A \rhd C[L], \Gamma) = Gr(A \rhd C[L])$ for all its clauses $A \rhd C[L]$.

An *SGGS-derivation* is a series of trails $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_j \vdash \ldots$, where $\Gamma_0 = \varepsilon$, and $\forall j,\ j > 0$, an SGGS-inference generates Γ_j from Γ_{j-1} and S. If $\bot \notin \Gamma$

and $I[\Gamma] \not\models S$, SGGS has *two ways to make progress*. If $\Gamma = dp(\Gamma)$, the trail is in order, but $I[\Gamma] \not\models C'$ for some $C' \in Gr(C)$ and $C \in S$. Then, SGGS applies *SGGS-extension* to generate from C and Γ a clause $A \rhd E$, such that E is an instance of C and $C' \in Gr(A \rhd E)$. If $\Gamma \neq dp(\Gamma)$, the trail needs repair: either there is a conflict, or there are intersections between selected literals to be removed by *SGGS-splitting*. The *SGGS-extension rules* specialize the *SGGS-extension scheme* ([14, Def. 12]) of which we give here the instance for I based on sign:

Definition 1. *Given input clause set S and trail Γ, if there is a clause $C \in S$ such that for all its I-true literals $L_1, \dots, L_n$ $(n \geq 0)$ there are clauses $B_1 \rhd D_1[M_1], \dots, B_n \rhd D_n[M_n]$ in $dp(\Gamma)$, such that literals $M_1, \dots, M_n$ are I-false, and $\forall j$, $1 \leqslant j \leqslant n$, $L_j\alpha = \neg M_j\alpha$ with simultaneous most general unifier (mgu) α, then* SGGS-extension *adds $A \rhd E = (\bigwedge_{j=1}^{n} B_j\alpha) \rhd C\alpha$ to Γ.*

SGGS-splitting decomposes a clause into instances to isolate and remove intersections between literals, and it is the only rule that introduces constraints. Splitting a ground clause is trivial and never done. Let $A \rhd C[L]$ be a clause where A is satisfiable. Roughly speaking (see [14, Sect. 3.2]), a *partition* of $A \rhd C[L]$ is a set $\{A_i \rhd C_i[L_i]\}_{i=1}^{n}$ such that $Gr(A \rhd C) = \bigcup_{i=1}^{n}\{Gr(A_i \rhd C_i)\}$ and the literals $A_i \rhd L_i$ are pairwise disjoint. Adding predicate symbol $\mathsf{Q} \subseteq s_1 \times s_2$ to Example 2, a partition of $[\mathsf{P}(\mathsf{f}(x), y)] \vee \mathsf{Q}(x, y)$ is $\{[\mathsf{P}(\mathsf{f}(x), \mathsf{b})] \vee \mathsf{Q}(x, \mathsf{b}),\ [\mathsf{P}(\mathsf{f}(x), \mathsf{f}(\mathsf{a}))] \vee \mathsf{Q}(x, \mathsf{f}(\mathsf{a}))\}$. Given trail clauses $A \rhd C[L]$ and $B \rhd D[M]$, a *splitting* of C by D, denoted *split*(C, D), is a partition of $A \rhd C[L]$ such that $at(Gr(A_j \rhd L_j))$ for some j is the intersection of $A \rhd L$ and $B \rhd M$, and all other $A_i \rhd L_i$ are disjoint from $B \rhd M$. *SGGS-splitting* replaces $A \rhd C[L]$ with *split*(C, D). Not all clauses in *split*(C, D) need to be kept for completeness (see [14, Sect. 4.2]).

Clause $A_n \rhd C_n[L_n]$ is *disposable*, if $I^p(\Gamma|_{n-1}) \models A_n \rhd C_n[L_n]$, and *SGGS-deletion* removes *all* disposable clauses from the trail. The following example shows that SGGS halts, if applied to the EPR set used to show that another semantically-guided method, hyperresolution, cannot decide EPR.

Example 3. The set S consists of four clauses ([18, Ex. 4.8] and [15, Ex. 3.17]):

$$\mathsf{P}(x, x, \mathsf{a}) \quad (i), \qquad \mathsf{P}(x, y, w) \vee \mathsf{P}(y, z, w) \vee \neg\mathsf{P}(x, z, w) \quad (ii),$$
$$\neg\mathsf{P}(x, x, \mathsf{b}) \quad (iii), \qquad \mathsf{P}(x, z, w) \vee \neg\mathsf{P}(x, y, w) \vee \neg\mathsf{P}(y, z, w) \quad (iv).$$

SGGS with all-negative initial interpretation I^- yields the following derivation:

$$\begin{array}{lll}
\Gamma_0 : \varepsilon \vdash \Gamma_1 : & [\mathsf{P}(x, x, \mathsf{a})] & \mathsf{extend}\ (i) \\
\vdash \Gamma_2 : & [\mathsf{P}(x, x, \mathsf{a})],\ \mathsf{P}(x, y, \mathsf{a}) \vee [\mathsf{P}(y, x, \mathsf{a})] \vee \neg\mathsf{P}(x, x, \mathsf{a}) & \mathsf{extend}\ (ii) \\
\vdash \Gamma_3 : & [\mathsf{P}(x, x, \mathsf{a})],\ \mathsf{P}(x, x, \mathsf{a}) \vee [\mathsf{P}(x, x, \mathsf{a})] \vee \neg\mathsf{P}(x, x, \mathsf{a}), & \\
 & y \neq x \rhd \mathsf{P}(x, y, \mathsf{a}) \vee [\mathsf{P}(y, x, \mathsf{a})] \vee \neg\mathsf{P}(x, x, \mathsf{a}) & \mathsf{split} \\
\vdash \Gamma_4 : & [\mathsf{P}(x, x, \mathsf{a})], & \\
 & y \neq x \rhd \mathsf{P}(x, y, \mathsf{a}) \vee [\mathsf{P}(y, x, \mathsf{a})] \vee \neg\mathsf{P}(x, x, \mathsf{a}) & \mathsf{delete}
\end{array}$$

Since $I^- \not\models P(x, x, \mathsf{a})$, SGGS-extension puts it on the trail. As $I[\Gamma_1]$ satisfies $P(x, x, \mathsf{a})$, but no other positive literal, $I[\Gamma_1] \not\models (ii)$. Thus, SGGS-extension unifies the third literal of clause (ii) with $[\mathsf{P}(x, x, \mathsf{a})]$ on the trail, producing Γ_2, where an I^--false (i.e., positive) literal is selected in the added clause (choosing the other makes no difference). As the selected literals intersect, the second clause gets split, yielding Γ_3. The second clause in Γ_3 is disposable and SGGS-deletion removes it. Since $I[\Gamma_4] \models S$, the derivation halts reporting satisfiable. In contrast, hyperresolution generates infinitely many clauses of the form $\mathsf{P}(x_1, x_2, \mathsf{a}) \vee \mathsf{P}(x_2, x_3, \mathsf{a}) \vee \cdots \vee \mathsf{P}(x_{n-1}, x_n, \mathsf{a}) \vee \mathsf{P}(x_n, x_1, \mathsf{a})$.

If a clause $A \rhd C[L]$ added by SGGS-extension is in conflict with $I[\Gamma]$ and $C[L]$ contains I-false literals, *SGGS-resolution explains the conflict* by resolving $A \rhd C[L]$ with a justification in $dp(\Gamma)$: the resolvent is also a conflict clause that replaces $A \rhd C[L]$. As SGGS-extension (see [14, Def. 19]) ensures that all I-false literals of a conflict clause can be resolved away, conflict explanation generates either $\perp$ or an I-all-true conflict clause $B \rhd D[M]$. The conflict represented by $B \rhd D[M]$ is *solved* by moving (*SGGS-move*) $B \rhd D[M]$ to the left of the clause in $dp(\Gamma)$ whose selected literal makes M uniformly false in $I[\Gamma]$: the effect is to *flip* M from being uniformly false to being an implied literal.

Fairness of an SGGS-derivation involves several properties: SGGS-deletion and other clause removals are applied eagerly; trivial splitting is avoided; progress is made whenever possible; every SGGS-extension generating a conflict clause is *bundled* with explanation and conflict-solving inferences to eliminate the conflict before new extensions occur; and inferences applying to shorter prefixes of the trail are never neglected in favor of others applying to longer prefixes (see [14, Defs. 32, 37, and 39]). The *limit* of a fair derivation $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_j \vdash \Gamma_{j+1} \vdash \ldots$ is the longest trail Γ_∞ such that $\forall i,\ i \leqslant |\Gamma_\infty|$, there is an n_i such that $\forall j$, $j \geqslant n_i$, if $|\Gamma_j| \geq i$ then $\Gamma_j|_i$ is equivalent to $\Gamma_\infty|_i$ (see [14, Def. 50]). In words, all prefixes of the trail stabilize eventually. Both derivation and Γ_∞ may be infinite, but if the derivation halts at stage k, $\Gamma_\infty = \Gamma_k$. The following results employ an *SGGS-suitable* (i.e., total and extending the size ordering, hence well-founded) ordering on ground atoms (see [14, Def. 16]) and a *convergence ordering* $>^c$ on SGGS-trails (see [14, Def. 46]):

- *Completeness*: For all input clause sets S, initial interpretations I, and fair SGGS-derivations, if S is satisfiable, $I[\Gamma_\infty] \models S$, and if S is unsatisfiable, $\perp \in \Gamma_k$ for some k (see [14, Thm. 9 and 11]).
- *Descending chain theorem*: A fair SGGS-derivation forms a descending chain $\Gamma_0 >^c \Gamma_1 >^c \ldots >^c \Gamma_j >^c \Gamma_{j+1} \ldots$ (see [14, Thm. 8]).
- *Finiteness of descending chains of length-bounded trails*: A chain $\Gamma_0 >^c \Gamma_1 >^c \ldots \Gamma_j >^c \Gamma_{j+1} \ldots$ where $\forall j,\ j \geq 0$, $|\Gamma_j| \leqslant n$, for some $n \geq 0$, is finite (see [14, Thm. 6 and Cor. 2]).

The gist of this paper is to find fragments where the length of SGGS-trails is bounded so that termination of fair derivations is guaranteed.

3 SGGS Decides the Stratified Fragment

A way to ensure termination is to restrict an inference engine to produce only terms or atoms from a finite set $\mathcal{B}$, usually called a *basis*. For SGGS, let $\mathcal{B}$ be a finite subset of the Herbrand base $\mathcal{A}$ of the input clause set S.

Definition 2. *An SGGS-trail* $\Gamma = A_1 \rhd C_1[L_1], \ldots, A_n \rhd C_n[L_n]$ is in $\mathcal{B}$ *if for all* i*,* $1 \leqslant i \leqslant n$*,* $at(Gr(A_i \rhd C_i)) \subseteq \mathcal{B}$*.*

An SGGS-derivation *is in* $\mathcal{B}$ if all its trails are.

Lemma 1. *If a fair SGGS-derivation* $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_j \vdash \Gamma_{j+1} \vdash \ldots$ *is in a finite basis* $\mathcal{B}$*, then for all* j*,* $j \geqslant 0$*,* $|\Gamma_j| \leqslant |\mathcal{B}|+1$*, and if the derivation halts at stage* k*,* $k \geqslant 0$*, then* $|\Gamma_k| \leqslant |\mathcal{B}|$*.*

Proof. SGGS cannot do worse than generating a ground trail where every atom in $\mathcal{B}$ appears selected with either sign: any trail with non-ground clauses will be shorter, because a non-ground clause covers many (possibly infinitely many) ground instances. By fairness, if the trail contains an intersection given by clauses $C[L]$ and $D[L]$, or $C[L]$ and $D[\neg L]$ with $L \in \mathcal{B}$, the clause on the right is either deleted eagerly by SGGS-deletion, or replaced with a resolvent by SGGS-resolution before SGGS-extension applies. Thus, there can be at most one such intersection, and the first claim follows. The second claim holds, because the intersection is removed by fairness prior to termination. □

By the descending chain theorem and the finiteness of descending chains of length-bounded trails, the following general result follows:

Theorem 1. *A fair SGGS-derivation in a finite basis is finite.*

In order to apply this result, we need to find fragments that admit a finite basis. We begin with the *stratified fragment*. A signature is *stratified*, if there is a well-founded ordering $<_s$ on sorts, and for all functions $f\colon s_1 \times \cdots \times s_n \rightarrow s$, it holds that $s <_s s_i$ for all $1 \leqslant i \leqslant n$ [1,25,36]. Thus, there are *no cycles over sorts* when applying functions. The signature from Example 2 is stratified with ordering $s_1 >_s s_2$. If a sentence over a stratified signature belongs to the $\exists^*\forall^*$ fragment, Skolemization only introduces constants and preserves stratification [25]. If there is only one sort, this fragment reduces to EPR, because stratification over a single sort implies that there are no function symbols. However, also stratified sentences with a prefix other than $\exists^*\forall^*$ can yield stratified clauses [33].

Example 4. Assume a stratified signature with sorts s_1 and s_2 such that $s_1 <_s s_2$, and symbols $\mathsf{f}\colon s_1 \rightarrow s_2$, and $\mathsf{P} \subseteq s_2 \times s_2$. The Skolemization of $\forall x \exists y.\, \mathsf{P}(\mathsf{f}(x), y)$ preserves stratification, as clause $\mathsf{P}(\mathsf{f}(x), \mathsf{g}(x))$ with Skolem symbol $\mathsf{g}\colon s_1 \rightarrow s_2$ is still stratified. On the other hand, the Skolemization of $\forall x \exists y.\, \mathsf{P}(\mathsf{f}(y), x)$ yields $\mathsf{P}(\mathsf{f}(\mathsf{g}(x)), x)$ with Skolem symbol $\mathsf{g}\colon s_2 \rightarrow s_1$, so that stratification is lost.

Given a set S of clauses whose signature is stratified, or stratified clause set for short, the Herbrand universe $\mathcal{H}$ and the Herbrand base $\mathcal{A}$ are *finite*, because stratification prevents building terms of unbounded depth [1]. Therefore, it suffices to pick $\mathcal{A}$ itself as the finite basis for Theorem 1.

Theorem 2. *Any fair SGGS-derivation from a stratified clause set* S *halts, is a refutation if* S *is unsatisfiable, and constructs a model of* S *if* S *is satisfiable.*

However, SGGS-derivations can get exponentially long.

Example 5. Consider the following clause set S_k describing a k-digits binary counter [38, Def. 2.4.10]. Let Q be a predicate symbol of arity k, and for all i, $1 \leqslant i \leqslant k$, let $\overline{\mathsf{0}}_i$, $\overline{\mathsf{1}}_i$, and $\overline{x}_i$ be i-tuples of $\mathsf{0}$'s, $\mathsf{1}$'s, and distinct variables $x_1, \ldots, x_i$, respectively. S_k consists of the $k+2$ clauses, for $1 \leqslant m \leqslant k$,

$$C_0\colon \mathsf{Q}(\overline{\mathsf{0}}_k) \quad C_m\colon \neg\mathsf{Q}(\overline{x}_m, \mathsf{0}, \overline{\mathsf{1}}_{k-m-1}) \vee \mathsf{Q}(\overline{x}_m, \mathsf{1}, \overline{\mathsf{0}}_{k-m-1}) \quad C_{k+1}\colon \neg\mathsf{Q}(\overline{\mathsf{1}}_k)$$

so that it is in EPR. Guided by I^-, SGGS generates a derivation

$$\begin{array}{lr}
\Gamma_0\colon \varepsilon \vdash \Gamma_1\colon [\mathsf{Q}(\overline{\mathsf{0}}_k)] & \mathsf{extend}\ (C_0)\\
\quad\vdash \Gamma_2\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{0}}_k) \vee [\mathsf{Q}(\overline{\mathsf{0}}_{k-1}, \mathsf{1})] & \mathsf{extend}\ (C_{k-1})\\
\quad\vdash \Gamma_3\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{0}}_{k-1}, \mathsf{1}) \vee [\mathsf{Q}(\overline{\mathsf{0}}_{k-2}, \mathsf{1}, \mathsf{0})] & \mathsf{extend}\ (C_{k-2})\\
\quad\vdash \Gamma_4\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{0}}_{k-2}, \mathsf{1}, \mathsf{0}) \vee [\mathsf{Q}(\overline{\mathsf{0}}_{k-2}, \mathsf{1}, \mathsf{1})] & \mathsf{extend}\ (C_{k-1})\\
\quad\ldots & \ldots\\
\quad\vdash \Gamma_{2^k-1}\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{1}}_{k-2}, \mathsf{0}, \mathsf{1}) \vee [\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0})] & \mathsf{extend}\ (C_{k-2})\\
\quad\vdash \Gamma_{2^k}\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0}) \vee [\mathsf{Q}(\overline{\mathsf{1}}_k)] & \mathsf{extend}\ (C_{k-1})\\
\quad\vdash \Gamma_{2^k+1}\colon \ldots, \neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0}) \vee [\mathsf{Q}(\overline{\mathsf{1}}_k)],\ [\neg\mathsf{Q}(\overline{\mathsf{1}}_k)] & \mathsf{extend}\ (C_{k+1})
\end{array}$$

that simulates binary counting by adding a clause in each of these 2^k+1 steps till a conflict emerges. Then it takes another 2^{k+1} steps to detect unsatisfiability:

$$\begin{array}{lr}
\vdash \Gamma_{2^k+2}\colon \ldots, [\neg\mathsf{Q}(\overline{\mathsf{1}}_k)],\ \neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0}) \vee [\mathsf{Q}(\overline{\mathsf{1}}_k)] & \mathsf{move}\\
\vdash \Gamma_{2^k+3}\colon \ldots, [\neg\mathsf{Q}(\overline{\mathsf{1}}_k)],\ [\neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0})] & \mathsf{resolve}\\
\vdash \Gamma_{2^k+2}\colon \ldots, [\neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0})],\ \neg\mathsf{Q}(\overline{\mathsf{1}}_{k-2}, \mathsf{0}, \mathsf{1}) \vee [\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0})],\ [\neg\mathsf{Q}(\overline{\mathsf{1}}_k)] & \mathsf{move}\\
\vdash \Gamma_{2^k+3}\colon \ldots, [\neg\mathsf{Q}(\overline{\mathsf{1}}_{k-1}, \mathsf{0})],\ [\neg\mathsf{Q}(\overline{\mathsf{1}}_{k-2}, \mathsf{0}, \mathsf{1})],\ [\neg\mathsf{Q}(\overline{\mathsf{1}}_k)] & \mathsf{resolve}\\
\ldots & \ldots\\
\vdash \Gamma_{2^{k+2}}\colon [\neg\mathsf{Q}(\overline{\mathsf{0}}_k)],\ [\mathsf{Q}(\overline{\mathsf{0}}_k)],\ \ldots & \mathsf{move}\\
\vdash \Gamma_{2^{k+2}+1}\colon \bot, \ldots & \mathsf{resolve}
\end{array}$$

Similar to positive resolution[1] [38, Thm. 2.4.12] or SCL [19], SGGS behaves exponentially, whereas resolution offers a refutation in $2k+1$ steps [38, Thm. 2.4.11], which shows that in EPR ground resolution (same as positive resolution for S_k) can do exponentially worse than resolution [34]. Encoding S_k in propositional logic requires exponentially many clauses, as each clause C_m, $1 \leqslant m \leqslant k$, has 2^m ground instances that need to be modeled as distinct propositional clauses. This indicates that generating instances is not good for this example.

4 SGGS Decides the Restrained Fragment

We begin with the notion of *ground-preserving* clause, which is convenient for sign-based semantic guidance.

[1] Every resolution step has a positive parent.

Definition 3. *A clause C is* positively ground-preserving *if $\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^-)$, and* negatively ground-preserving *if $\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^+)$. A set of clauses is* positively/negatively ground-preserving *if all its clauses are, and* ground-preserving *if it is positively or negatively ground-preserving.*

For example, $\neg\mathsf{P}(x, y, z) \vee \mathsf{Q}(y) \vee \mathsf{Q}(\mathsf{f}(z))$ and $\neg\mathsf{Q}(x) \vee \neg\mathsf{Q}(y)$ are positively ground-preserving, while the clauses in Example 5 are both positively and negatively ground-preserving. We say that I is *suitable* for a ground-preserving set S if either I is I^- and S is positively ground-preserving, or I is I^+ and S is negatively ground-preserving. If S is positively ground-preserving, its positive clauses are ground, positive hyperresolution only generates ground clauses ([12], Lem. 3), and similarly for the negative variant. We show that this also holds for SGGS.

Lemma 2. *If the input clause set S is ground-preserving and the initial interpretation is suitable for S, any fair SGGS-derivation from S is ground.*

Proof. We consider S positively ground-preserving and I^- (for the dual case one exchanges the signs). The proof is by induction on the length n of the derivation. The base case ($n = 0$) is vacuously true. The induction hypothesis is that the claim holds for a derivation of length n producing trail Γ. Let $\Gamma \vdash \Gamma'$ be the (n+1)-th step. Since Γ is ground, $\Gamma \vdash \Gamma'$ cannot be a splitting step, because any splitting of a ground clause yields the clause itself, and fairness excludes such trivial splittings ([14, Defs. 32, 47, and 49]). If $\Gamma \vdash \Gamma'$ is an SGGS-resolution step, it is a ground resolution step, and also Γ' is ground. If $\Gamma \vdash \Gamma'$ is an SGGS-extension step, it adds an instance $C\alpha$ of a clause $C \in S$, where α is the simultaneous mgu of *all* I^--true (i.e., negative) literals $L_1, \dots, L_n$ in C with as many I^--false (i.e., positive) selected literals $M_1, \dots, M_n$ in Γ (see Def. 1). Since Γ is ground by induction hypothesis, the clauses containing $M_1, \dots, M_n$ are ground and do not have constraints. Thus, $L_1\alpha, \dots, L_n\alpha$ are also ground. The I^--false literals of $C\alpha$ are ground, because C is positively ground-preserving (i.e., $\mathcal{V}ar(C) \subseteq \mathcal{V}ar(C^-)$), so that all its variables get grounded by α. Hence $C\alpha$ and Γ' are ground. $\square$

The next example illustrates Lemma 2 and gives the intuition for restrainedness.

Example 6. Assume a positively ground-preserving set S which includes:

$$\mathsf{P}(\mathsf{s}^{10}(0), \mathsf{s}^9(0)) \quad (i), \quad \neg\mathsf{P}(\mathsf{s}(\mathsf{s}(x)), y) \vee \mathsf{P}(x, \mathsf{s}(y)) \quad (ii), \quad \neg\mathsf{P}(\mathsf{s}(0), 0) \quad (iii),$$

and I^- is the initial interpretation. SGGS starts with an extension that puts the positive clause $\mathsf{P}(10, 9)$ on the trail, where we write n for $\mathsf{s}^n(0)$. Subsequent extensions unify the negative literal in clause (ii) with some positive ground literal on the trail, so that new literals in added clauses are positive:

$$\begin{aligned}
\Gamma_0 : \varepsilon \vdash\ & \Gamma_1 : [\mathsf{P}(10, 9)] \\
\vdash\ & \Gamma_2 : [\mathsf{P}(10, 9)],\ \neg\mathsf{P}(10, 9) \vee [\mathsf{P}(8, 10)] \\
\vdash\ & \Gamma_3 : [\mathsf{P}(10, 9)],\ \neg\mathsf{P}(10, 9) \vee [\mathsf{P}(8, 10)],\ \neg\mathsf{P}(8, 10) \vee [\mathsf{P}(6, 11)].
\end{aligned}$$

The positive literals have decreasing number of symbols, matching the fact that $\mathsf{P}(\mathsf{s}(\mathsf{s}(x)), y) \succ \mathsf{P}(x, \mathsf{s}(y))$ in (ii) for $\succ$ any lexicographic path ordering (LPO).

This suggests to strengthen ground-preservingness with an ordering (unrelated to the SGGS-suitable ordering of Sect. 2) to get a finite basis.

Definition 4. *A quasi-ordering $\succeq$ on terms and atoms is* restraining*, if (i) it is stable under substitution, (ii) the strict ordering $\succ = \succeq \setminus \preceq$ is well-founded, and (iii) the equivalence $\approx = \succeq \cap \preceq$ has finite equivalence classes.*

Note that Condition (i) implies that $\succ$ and $\approx$ are stable under substitution. From now on, $\succeq$ is a restraining quasi-ordering. Let $\mathcal{A}_S$ be the set of ground atoms occurring in S, and $\mathcal{A}_S^{\preceq}$ the subset of the Herbrand base $\mathcal{A}$ of ground atoms upper-bounded by $\mathcal{A}_S$, so $\mathcal{A}_S^{\preceq} = \{L : L \in \mathcal{A}, \exists M \in \mathcal{A}_S \text{ with } M \succeq L\}$. By Conditions (ii) and (iii) in Definition 4, $\mathcal{A}_S^{\preceq}$ is finite and thus can serve as basis.

Definition 5. *A clause C is* (strictly) positively restrained *if it is positively ground preserving, and for all non-ground literals $L \in C^+$ there is a literal $M \in C^-$ such that $at(M) \succeq at(L)$ ($at(M) \succ at(L)$). A set of clauses is* positively restrained *if all its clauses are.*

Negatively restrained clauses and clause sets are defined similarly, and a set of clauses is *restrained* if it is positively or negatively restrained. The set in Example 6 is strictly positively restrained. We see next the role of the *quasi*-ordering.

Example 7. Problem PLA030-1 in TPTP is neither stratified, nor monadic, nor guarded. Its clause $\mathsf{differ}(x, y) \vee \neg\mathsf{differ}(y, x)$ cannot be shown *strictly* restrained. Let $\succ_{\mathsf{acrpo}}$ be an AC-compatible [40] recursive path ordering with differ as an AC-symbol, meaning associative-commutative. The quasi-ordering $\succeq_{\mathsf{acrpo}}$, built from $\succ_{\mathsf{acrpo}}$ and the AC-equivalence $\approx_{\mathsf{AC}}$ that has finite equivalence classes, satisfies $\mathsf{differ}(x, y) \succeq_{\mathsf{acrpo}} \mathsf{differ}(y, x)$, and shows that PLA030-1 is negatively restrained.

Restrainedness is undecidable in general, but decidable for fixed, suitable orderings. If S is restrained, a fair SGGS-derivation will be in $\mathcal{A}_S^{\preceq}$.

Lemma 3. *If the input clause set S is restrained and the initial interpretation is suitable for S, any fair SGGS-derivation from S is in $\mathcal{A}_S^{\preceq}$.*

Proof. We consider S positively ground-preserving and I^- (for the dual case one exchanges the signs). Since the set is restrained hence ground-preserving, the derivation is ground by Lemma 2 (†). The proof is by induction on the length n of the derivation, and it follows the same pattern as that of Lemma 2. Let $\Gamma \vdash \Gamma'$ be the $(n{+}1)$-th step. By induction hypothesis, Γ is in $\mathcal{A}_S^{\preceq}$. If $\Gamma \vdash \Gamma'$ is an SGGS-resolution step, it is a ground resolution step which does not generate new atoms, and also Γ' is in $\mathcal{A}_S^{\preceq}$. If $\Gamma \vdash \Gamma'$ is an SGGS-extension step, it adds an instance $C\alpha$ of a clause $C \in S$, where α is the simultaneous mgu of all I^--true (i.e., negative) literals $\neg L_1, \dots, \neg L_n$ in C with as many I^--false (i.e., positive) selected literals $M_1, \dots, M_n$ in Γ. The literals $M_1, \dots, M_n$ are ground by (†), and by induction hypothesis they are in $\mathcal{A}_S^{\preceq}$. We have to show $at(C\alpha) \subseteq \mathcal{A}_S^{\preceq}$.

– For the negative literals $\neg L_1\alpha, \dots, \neg L_n\alpha$ we have $L_i\alpha = M_i\alpha = M_i \in \mathcal{A}_S^{\preceq}$.
– Let L be a literal in C^+. If L is ground, then $L\alpha = L \in \mathcal{A}_S \subseteq \mathcal{A}_S^{\preceq}$. If L is not ground, by positive restrainedness there exists a $\neg L_i$, $1 \leqslant i \leqslant n$, such that $L_i \succeq L$. By stability, $L_i\alpha \succeq L\alpha$. Since for all i, $1 \leqslant i \leqslant n$, $M_i \in \mathcal{A}_S^{\preceq}$ and $M_i = M_i\alpha = L_i\alpha \succeq L\alpha$, we have $L\alpha \in \mathcal{A}_S^{\preceq}$. □

Therefore, as $\mathcal{A}_S^{\preceq}$ is finite, Theorem 1 applies.

Theorem 3. *Any fair SGGS-derivation from a restrained clause set S with a suitable initial interpretation for S halts, is a refutation if S is unsatisfiable, and constructs a model of S if S is satisfiable.*

Restrainedness also makes it possible to derive an upper bound on the cardinality of a single-sorted model, defined as the cardinality of its domain. Let $\mathcal{H}_S^{\preceq}$ be the set $\mathcal{H}_S^{\preceq} = \{t : t \text{ is a strict subterm of } L \text{ for some } L \in \mathcal{A}_S^{\preceq}\}$.

Theorem 4. *A restrained satisfiable clause set S has a model of cardinality at most $|\mathcal{H}_S^{\preceq}| + 1$ that can be extracted from the limit of any fair SGGS-derivation from S with suitable initial interpretation I.*

Proof. By Theorem 3 the derivation halts with some trail Γ, and by Lemmas 2 and 3, Γ contains only ground clauses whose atoms are in $\mathcal{A}_S^{\preceq}$. Since SGGS is model complete, $I[\Gamma] \models S$. Consider the following interpretation J with domain $\mathcal{H}_S^{\preceq} \uplus \{u\}$, where u is a new constant symbol: for every constant symbol c we set $c^J = c$ if $c \in \mathcal{H}_S^{\preceq}$, and $c^J = u$ otherwise. For every n-ary ($n \geqslant 1$) function symbol f, we set $f^J(t_1, \dots, t_n) = f(t_1^J, \dots, t_n^J)$ if $f(t_1, \dots, t_n) \in \mathcal{H}_S^{\preceq}$, and $f^J(t_1, \dots, t_n) = u$ otherwise. For every predicate symbol P, $(t_1, \dots, t_n) \in P^J$ if $I[\Gamma] \models P(t_1, \dots, t_n)$. Note that J is well-defined because if $f(t_1, \dots, t_n) \in \mathcal{H}_S^{\preceq}$ then $t_1, \dots, t_n$ are also, hence all terms are interpreted in $\mathcal{H}_S^{\preceq} \uplus \{u\}$. As J agrees with $I[\Gamma]$ on all atoms, $J \models S$, and it has cardinality $|\mathcal{H}_S^{\preceq}| + 1$ by construction. □

Therefore the restrained fragment also enjoys the *small model property.*

Example 8. The satisfiable clause set S (PUZ054-1 in TPTP) extending Example 6:

$$\begin{array}{lll}
\mathsf{P}(\mathsf{s}^{10}(\mathsf{0}), \mathsf{s}^9(\mathsf{0})), & \neg\mathsf{P}(\mathsf{s}(\mathsf{s}(x)), y) \vee \mathsf{P}(x, \mathsf{s}(y)), & \neg\mathsf{P}(x, \mathsf{s}(\mathsf{s}(y))) \vee \mathsf{P}(x, \mathsf{s}(y)), \\
\neg\mathsf{P}(\mathsf{s}(\mathsf{0}), \mathsf{0}), & \neg\mathsf{P}(\mathsf{s}(x), \mathsf{s}(y)) \vee \mathsf{P}(\mathsf{s}(x), y), &
\end{array}$$

is neither in EPR, nor in FO^2, nor in the monadic class. However, it can be shown strictly positively restrained by a Knuth-Bendix ordering (KBO) $\succ$ with empty precedence and weights $w(\mathsf{P}) = 0$ and $w_0 = w(\mathsf{s}) = w(\mathsf{0}) = 1$, where w_0 is the weight of variables. The largest atom in $\mathcal{A}_S = \{\mathsf{P}(\mathsf{s}^{10}(\mathsf{0}), \mathsf{s}^9(\mathsf{0})),\ \mathsf{P}(\mathsf{s}(\mathsf{0}), \mathsf{0})\}$ has weight $w(\mathsf{P}(\mathsf{s}^{10}(\mathsf{0}), \mathsf{s}^9(\mathsf{0}))) = 21$. No atom L of the form $\mathsf{P}(\mathsf{s}^n(\mathsf{0}), \mathsf{s}^m(\mathsf{0}))$ in $\mathcal{A}_S^{\preceq}$ can have a subterm $\mathsf{s}^k(\mathsf{0})$ with $k > 19$, because otherwise $w(L) > w(\mathsf{P}(\mathsf{s}^{10}(\mathsf{0}), \mathsf{s}^9(\mathsf{0})))$. Therefore, we have $\mathcal{H}_S^{\preceq} = \{\mathsf{s}^i(\mathsf{0}) \, : \, 0 \leqslant i \leqslant 19\}$ and there is a model of cardinality at most 21 by Theorem 4.

We next consider PVD [17]. Let depth(C) be the maximum depth of an atom in clause C, and depth$_x(C)$ the maximum occurrence depth in C of $x \in \mathcal{V}ar(C)$.

Definition 6. *A clause set S is in PVD if every clause $C \in S$ is positively ground-preserving and $\forall x \in Var(C^+)$ it holds that $depth_x(C^+) \leqslant depth_x(C^-)$.*

Let $\mathcal{A}_d$ be the subset of the Herbrand base $\mathcal{A}$ containing all ground atoms whose depth does not exceed d.

Lemma 4. *If the input clause set S is in PVD and has maximal depth d then any fair SGGS-derivation from S using I^- is in $\mathcal{A}_d$.*

Proof. Since S is in PVD and hence ground-preserving, by Lemma 2 the derivation is ground (†). The proof is by induction on the length n of the derivation, and follows that of Lemma 3 except for the extension case in the inductive step. Let $\Gamma \vdash \Gamma'$ be an SGGS-extension step as described in the proof of Lemma 3. The positive literals $M_1, \ldots, M_n$ are ground by (†) and by induction hypothesis in $\mathcal{A}_d$. Since $L_i\alpha = M_i\alpha = M_i$ for all i, $1 \leqslant i \leqslant n$, the atoms $L_i\alpha$ in $C^-\alpha$ are ground and in $\mathcal{A}_d$. Let L be a literal in C^+. Since S is in PVD, every variable in L occurs at a lower or equal depth in some L_i $(1 \leqslant i \leqslant n)$. Therefore, $L\alpha$ is ground and its depth cannot exceed that of $L_i\alpha$, so that $L\alpha \in \mathcal{A}_d$. □

Since $\mathcal{A}_d$ is a finite basis, termination follows from Theorem 1.

Theorem 5. *Any fair SGGS-derivation using I^- from a PVD set S halts, is a refutation if S is unsatisfiable, and constructs a model of S if S is satisfiable.*

5 Ordered Resolution Decides the Restrained Fragment

In this section we work with the positively restrained fragment; the case for the negatively restrained one is symmetric. Let $>$ be a stable and well-founded ordering on literals, such that positive literals are maximal only in positive clauses, so that ordered resolution is positive ordered resolution. In this way, the ordering embeds the suitable sign-based semantic guidance for the positively restrained fragment. The ordering $>$ could be the extension of the restraining ordering $\succ$ (see Definition 5) to literals, but does not have to be. We use the following notations [18]: $Res_>(S)$ denotes the set of ordered resolvents generated from parents in S; $R^0_>(S) = S$, $R^{k+1}_>(S) = R^k_>(S) \cup Res_>(R^k_>(S))$, and $R^*_>(S) = \bigcup_{k \geqslant 0} R^k_>(S)$. We begin with an auxiliary lemma.

Lemma 5. *If S is positively restrained, then for all $C \in R^*_>(S)$, for all $L \in C^+$ either (i) $L \in \mathcal{A}_S^{\preceq}$, or (ii) $at(M) \succeq at(L)$ for some $M \in C^-$.*

Proof. The proof is by induction on the stage k of the construction of $R^*_>(S)$. For $k = 0$, the clauses in $R^0_>(S) = S$ satisfy the claim by the definitions of restrainedness, $\mathcal{A}_S$, and $\mathcal{A}_S^{\preceq}$. The induction hypothesis is that all clauses in $R^k_>(S)$ satisfy the claim. For the inductive step, let $C\sigma \vee D\sigma$ be a resolvent in $Res_>(R^k_>(S))$ generated from parents $\neg L \vee C$ and $L' \vee D$, where $\neg L\sigma$ and $L'\sigma$ are $>$-maximal literals, and $L\sigma = L'\sigma$ for mgu σ. The clause $L' \vee D$ must be positive, otherwise $L'\sigma$ cannot be $>$-maximal. Since $L' \vee D \in R^k_>(S)$, by induction hypothesis $at(L' \vee D) \subseteq \mathcal{A}_S^{\preceq}$ (†), which means $L' \vee D$ is ground, $(L' \vee D)\sigma = L' \vee D$, and all atoms in $D\sigma$ are in $\mathcal{A}_S^{\preceq}$. For the positive literals in $C\sigma$, let $M\sigma$ be one of them, so $M \in C^+$. Since $\neg L \vee C$ is in $R^k_>(S)$, by induction hypothesis, either (i) $M \in \mathcal{A}_S^{\preceq}$, or (ii) $M' \succeq M$ for some negative literal $\neg M'$ in $\neg L \vee C$. In case (i), M is ground, $M\sigma = M$, and $M\sigma \in \mathcal{A}_S^{\preceq}$. In case (ii), if $\neg M'$ occurs in C then $\neg M'\sigma \in C\sigma$, and $M'\sigma \succeq M\sigma$ holds by stability, so that the claim holds. Otherwise, $\neg M'$ is the resolved-upon literal $\neg L$ with $L\sigma = L'\sigma$. Thus, $L = M' \succeq M$, which implies $L\sigma \succeq M\sigma$ by stability, and since L' is ground, $L'\sigma = L'$. By (†), $L' \in \mathcal{A}_S^{\preceq}$. Since $L' = L'\sigma = L\sigma \succeq M\sigma$, we have $M\sigma \in \mathcal{A}_S^{\preceq}$ by definition of $\mathcal{A}_S^{\preceq}$. □

Theorem 6. *Any fair ordered resolution derivation using $>$ from a restrained clause set S terminates, and is a refutation if S is unsatisfiable.*

Proof. Let S be positively restrained. We show that $R^*_>(S)$ is finite. The claim then follows from refutational completeness of ordered resolution. We first show the following (†): for every ordered resolvent $C\sigma \vee D\sigma$ in $R^*_>(S)$ from parents $\neg L \vee C$ and $L' \vee D$ and mgu σ, $L' \vee D$ is ground and positive, and $C\sigma \vee D\sigma$ has strictly fewer negative literals than $\neg L \vee C$. Indeed, by the definition of ordered resolution, $\neg L\sigma$ and $L'\sigma$ are $>$-maximal in $(\neg L \vee C)\sigma$ and $(L' \vee D)\sigma$, respectively. The clause $L' \vee D$ must be positive, otherwise $L'\sigma$ cannot be $>$-maximal. By Lemma 5, every positive literal in a positive clause is in $\mathcal{A}_S^{\preceq}$, and therefore it is ground. Thus, in the resolvent $C\sigma \vee D\sigma$ the literals in $D\sigma$ are positive, hence $C\sigma \vee D\sigma$ has fewer negative literals than $\neg L \vee C$.

Now suppose that $R^*_>(S)$ is infinite. Observation (†) reveals that the number of negative literals decreases with every resolution step. Hence, if $R^*_>(S)$ is infinite, it must contain infinitely many positive clauses. By Lemma 5, these positive clauses are ground, and all their atoms are in the finite basis $\mathcal{A}_S^{\preceq}$. As repeated literals in ground clauses disappear by merging, the number of ground clauses that can be built from $\mathcal{A}_S^{\preceq}$ is finite. This contradicts $R^*_>(S)$ being infinite. □

This result extends to other positive resolution strategies.

Corollary 1. *Hyperresolution and $>$-ordered resolution with negative selection decide the positively restrained fragment.*

The next example shows that SGGS can be exponentially more efficient than these saturation-based resolution strategies because it is model-based.

Example 9. Consider the following parametric clause set S_n consisting of $n+1$ clauses, using $i{+}1$-ary predicates P_i and constants c_i, for all i, $0 \leqslant i \leqslant n$:

$$\begin{array}{lr}
\mathsf{P}_0(\mathsf{c}_0) \vee \mathsf{P}_0(\mathsf{c}_1) \vee \cdots \vee \mathsf{P}_0(\mathsf{c}_n) & (C_0),\\
\neg\mathsf{P}_0(x_1) \vee \mathsf{P}_1(x_1, \mathsf{c}_0) \vee \mathsf{P}_1(x_1, \mathsf{c}_1) \vee \cdots \vee \mathsf{P}_1(x_1, \mathsf{c}_n) & (C_1),\\
\neg\mathsf{P}_1(x_1, x_2) \vee \mathsf{P}_2(x_1, x_2, \mathsf{c}_0) \vee \cdots \vee \mathsf{P}_2(x_1, x_2, \mathsf{c}_n) & (C_2),\\
\ldots & \ldots\\
\neg\mathsf{P}_{n-1}(x_1, \ldots, x_n) \vee \mathsf{P}_n(x_1, \ldots, x_n, \mathsf{c}_0) \vee \cdots \vee \mathsf{P}_n(x_1, \ldots, x_n, \mathsf{c}_n) & (C_n).
\end{array}$$

The set S_n is positively restrained by an LPO with precedence $\mathsf{P}_0 > \cdots > \mathsf{P}_n > \mathsf{c}_i$ for all i, $0 \leqslant i \leqslant n$. SGGS with I^- simply selects one positive literal per clause and detects satisfiability after $n+1$ SGGS-extension steps, producing for instance the model where $\mathsf{P}_0(\mathsf{c}_0), \mathsf{P}_1(\mathsf{c}_0, \mathsf{c}_0), \ldots, \mathsf{P}_n(\mathsf{c}_0, \ldots, \mathsf{c}_0)$ are true and all other atoms are false. A saturation by any of the above positive resolution strategies produces exponentially many clauses, because for all i, $0 \leqslant i \leqslant n$, all n positive literals in C_i unify with the negative literal in C_{i+1}, generating n^{i+1} positive clauses, so that the clause count is given by $\sum_{k=1}^{n} n^k$.

6 Experiments

We begin by reducing positive restrainedness (the negative case is similar) to termination of rewrite systems, so that termination tools can yield a partial practical test. Since restrainedness employs a quasi-ordering, we consider unsorted rewrite systems $\mathcal{R}$ and $\mathcal{E}$, and *rewriting modulo* $\rightarrow_{\mathcal{R}/\mathcal{E}}$ defined by $\leftrightarrow^*_{\mathcal{E}} \circ \rightarrow_{\mathcal{R}} \circ \leftrightarrow^*_{\mathcal{E}}$.

Definition 7. *Given a clause set S, a pair of rewrite systems $(\mathcal{R}_S, \mathcal{E}_S)$ is* positively restraining *for S if for all clauses $C \in S$ and all non-ground literals $L \in C^+$, there is a rule $at(M) \rightarrow at(L)$ in $\mathcal{R}_S \cup \mathcal{E}_S$ for some literal $M \in C^-$.*

Let $(\mathcal{R}_S, \mathcal{E}_S)$ be positively restraining for S. For instance, for Example 8, $\mathcal{R}_S$ will consist of $\mathsf{P}(\mathsf{s}(\mathsf{s}(x)), y) \rightarrow \mathsf{P}(x, \mathsf{s}(y))$, $\mathsf{P}(x, \mathsf{s}(\mathsf{s}(y))) \rightarrow \mathsf{P}(x, \mathsf{s}(y))$, and $\mathsf{P}(\mathsf{s}(x), \mathsf{s}(y)) \rightarrow \mathsf{P}(\mathsf{s}(x), y)$, while $\mathcal{E}_S = \emptyset$. A common situation is that $\mathcal{E}_S$ has permutative rules, as in Example 7 where $\mathsf{differ}(x, y) \rightarrow \mathsf{differ}(y, x)$ will be in $\mathcal{E}_S$.

Lemma 6. *(1) If $\rightarrow_{\mathcal{R}_S}$ is terminating and $\mathcal{E}_S = \emptyset$, clause set S is strictly positively restrained. (2) If $\rightarrow_{\mathcal{R}_S/\mathcal{E}_S}$ is 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.*

Proof. (1) Since $\rightarrow_{\mathcal{R}_S}$ is terminating, for all $t \rightarrow u$ in $\mathcal{R}_S$, $\mathcal{V}ar(u) \subseteq \mathcal{V}ar(t)$, so that S is positively ground-preserving. S is strictly positively restrained by the quasi-ordering $\rightarrow^*_{\mathcal{R}_S}$: indeed, $\rightarrow^*_{\mathcal{R}_S}$ is stable, $\rightarrow^+_{\mathcal{R}_S}$ is well-founded, and the equivalence classes of $\rightarrow^*_{\mathcal{R}_S} \cap {}_{\mathcal{R}_S}^{*}\!\!\leftarrow$, which is identity, are finite. (2) Since $\rightarrow_{\mathcal{R}_S/\mathcal{E}_S}$ is terminating, for all $t \rightarrow u$ in $\mathcal{R}_S$, $\mathcal{V}ar(u) \subseteq \mathcal{V}ar(t)$, and for all $t \rightarrow u$ in $\mathcal{E}_S$, $\mathcal{V}ar(u) = \mathcal{V}ar(t)$ by hypothesis, so that S is positively ground-preserving. S is positively restrained by $\rightarrow^*_{\mathcal{R}_S/\mathcal{E}_S}$: indeed, $\rightarrow^*_{\mathcal{R}_S/\mathcal{E}_S}$ is stable, $\rightarrow^+_{\mathcal{R}_S/\mathcal{E}_S}$ is well-founded, and the equivalence classes of $\leftrightarrow^*_{\mathcal{E}}$ are finite by hypothesis. $\square$

For the experiments, given a clause set S, a script named StoR generates rewrite systems $\mathcal{R}_S$ and $\mathcal{E}_S$ that can be fed to a termination tool. The source of clause sets is TPTP 7.2.0 and the termination tool is $\mathsf{T_TT_2}$ [27]. All problems in the FOF category are transformed into conjunctive normal form, excluding those with equality (whether sets with equality can be restrained is a topic for future work). Besides 1,539 inputs where either StoR or $\mathsf{T_TT_2}$ timed out, out of the remaining 3,462 problems, $\mathsf{T_TT_2}$ found 313 restrained ones. For those still undetermined, we tested whether it is sufficient to flip the sign of all literals with a certain predicate to get a restrained problem, which succeeded in 36 cases, for a total of 349 restrained problems. Of these, 277 are positively restrained, 181 negatively restrained, and 109 are both; 74 are ground, 232 are PVD, 277 are stratified, 252 are EPR, 169 are monadic, 204 are FO^2, 209 are guarded, but 43 problems do not fall in any of these classes, and therefore, to the best of our knowledge, they are proved decidable here for the first time. The average TPTP rating of the 349 problems is 0.1, and that of the 43 problems is 0.015, where 0.1 means that the problem can be solved by most provers. However, the group of 349 includes hard problems such as instances of the binary counter problem in Example 5 (MSC015-1.n), and Rubik's cube problems (e.g., PUZ052-1). For example, MSC015-1.030 is restrained and has rating 1.00, that is, no theorem prover could solve it so far within the timeout allowed in the CASC competition.

Example 10. Problem HWV036-2 (cf. Example 1) is a set of axioms with no ground atoms, so that $\mathcal{H}_S^{\preceq}$ is empty, and the model constructed according to Theorem 4 is trivial. Several other problems combine this set with ground clauses to prove theorems from those axioms. For example, HWV008-2.002 adds 23 ground clauses. As we found a terminating positively restraining rewrite system for HWV008-2.002, this problem, as well as HWV036-2, is strictly restrained.

An SGGS prototype named Koala was built reusing code for basic data structures, term indexing, and type inference from iProver [24,26]. In Koala, the SGGS-trail is represented as a list of constrained clauses, with constraints maintained in *standardized form* (see Sect. 2 and [14, Sect. 7]), and selected literals stored in a *discrimination tree*, since SGGS-extension requires to find selected literals that unify with literals in an input clause (cf. Definition 1). Koala takes sorts into account when checking satisfiability of constraints (e.g., Example 2), and implements a *fair search plan* which ensures that all derivations are fair (see Sect. 2). The *SGGS-suitable ordering* is a KBO with built-in precedence, $w_0 = 1$, and weight 1 for all function and predicate symbols in order to extend the size ordering. Koala also sorts by this ordering the clauses in a splitting, according to the SGGS notion of *preferred clause* in a splitting [14, Def. 22]).

Koala picks either I^- or I^+ as initial interpretation, based on whether the problem is positively or negatively ground-preserving, which overapproximates positively or negatively restrained. Koala implements the above mentioned sign-flipping test to obtain more ground-preserving sets. In order to recognize stratified input problems, one can compute the sort dependency graph and check for acyclicity [25], or let Koala apply type inference. For all experiments with Koala, the time-out was 300 sec of wall clock time. Out of the above 349 restrained problems, Koala shows that 50 are satisfiable and 283 unsatisfiable. Of 351 PVD problems, not including the 232 ones in the set of 349 restrained problems, Koala discovers 232 unsatisfiable and 76 satisfiable instances. Of the 1,246 stratified problems found in the FOF category by the above acyclicity test, Koala solves 643 unsatisfiable and 277 satisfiable instances. The list of the 349 restrained sets with their properties and restraining rewrite systems, as well as detailed results of the experiments with Koala are available.[2]

7 Discussion

SGGS [13,14] is an attractive candidate for decision procedures, because it is conflict-driven and it builds models. In this paper, we showed that if the generated clauses are in a *finite basis*, SGGS is guaranteed to terminate, and we instantiated this result to yield SGGS decision procedures for the *stratified fragment*, *PVD*, and the newly introduced *restrained fragment*, all without equality.

While also Inst-Gen [24] and the model evolution calculus (MEC) [7] decide the stratified fragment [25], this is not the case for the restrained fragment: for instance, if Inst-Gen uses an unfortunate literal selection in Example 8, it does not halt. SGGS avoids this phenomenon thanks to semantic guidance. Since MEC starts with I^+ as candidate model, it may not terminate on satisfiable negatively restrained sets such as Example 7. Indeed, E-Darwin [6] does not halt on this example that Koala solves in a few seconds.

However, it is generally difficult to tame a refutationally complete first-order inference system to yield decision procedures. For instance, SGGS does not halt given the following set of clauses that belongs to several decidable fragments.[3]

Example 11. The following set is in the Ackermann, monadic, and FO^2 classes:

$$\mathsf{P}(\mathsf{0}) \quad (i) \qquad \mathsf{P}(x) \vee \mathsf{P}(\mathsf{f}(x)) \quad (ii) \qquad \neg\mathsf{P}(x) \vee \neg\mathsf{P}(\mathsf{f}(x)) \quad (iii),$$

where membership in the Ackermann class stems from the Skolemization of $\exists v \forall x \exists y. P(v) \wedge (P(x) \vee P(y)) \wedge (\neg P(x) \vee \neg P(y))$. It has the finite model property, as witnessed by model $\mathcal{I}$ with domain $\{0, 1\}$, $\mathsf{f}^{\mathcal{I}}(x) = 1 - x$, $\mathsf{0}^{\mathcal{I}} = 0$, and $\mathsf{P}^{\mathcal{I}} = \{0\}$. With I^-, SGGS performs an infinite series of non-conflicting extensions:

$$\begin{array}{lll}
\varepsilon & \vdash [\mathsf{P}(\mathsf{0})] & \mathsf{extend}\ (i) \\
 & \vdash [\mathsf{P}(\mathsf{0})],\ \neg\mathsf{P}(\mathsf{0}) \vee [\neg\mathsf{P}(\mathsf{f}(\mathsf{0}))] & \mathsf{extend}\ (iii) \\
 & \vdash [\mathsf{P}(\mathsf{0})],\ \neg\mathsf{P}(\mathsf{0}) \vee [\neg\mathsf{P}(\mathsf{f}(\mathsf{0}))],\ \mathsf{P}(\mathsf{f}(\mathsf{0})) \vee [\mathsf{P}(\mathsf{f}(\mathsf{f}(\mathsf{0})))] & \mathsf{extend}\ (ii) \\
 & \vdash \ldots &
\end{array}$$

It can be shown that SGGS does not terminate with I^+ either. Inst-Gen may terminate if appropriate candidate models are constructed, but not in general. Ordered resolution terminates if the ordering satisfies $\neg\mathsf{P}(\mathsf{f}(x)) > \neg\mathsf{P}(x)$.

[2] http://profs.scienze.univr.it/winkler/sggsdp/.

[3] Renate A. Schmidt and Marco Voigt suggested this example to the first author.

The stratified fragment was presented as the first of three fragments of increasing expressivity, named St_0, St_1, and St_2, and suitable to capture Alloy specifications [1]. If the generic predicate symbols P of Example 4 is replaced with equality for sort s_2, the second sentence of Example 4 becomes $\forall x \exists y.\, f(y) \simeq x$, which states that x is in the image of f: this formula separates St_0 and St_1, since it is allowed in St_1, but not in St_0. The LISBQ logic for the verification of unbounded data structures [29] can be translated into St_0 with equality [1].

The *restrained fragment* is defined starting from a notion of *ground-preservingness*, which is convenient for positive or negative strategies. Negative ground-preservingness was used for Horn theories with equality [28] and [10, Sect. 5.2]. Positive ground-preservingness, also named *range restrictedness* [15], is implicit in PVD and recent extensions [30]. Unlike these settings, the restrained fragment does not limit literal depth. Positive ground-preservingness was used to show that DPLL(Γ+$\mathcal{T}$), where Γ is an inference system including hyperresolution, superposition with negative selection, and simplification, decides *essentially finite* (only one monadic function f with finite range) and positively ground-preserving axiomatizations, provided that *speculative axioms* $f^j(x) \simeq f^k(x)$ $(j > k)$ are tried for increasing values of j and k ([12], Thm. 7 and Lem. 3). Simplification applies the speculative axioms to limit the depth of generated terms. Without this feature, it is not surprising that SGGS does not halt.

Example 12. Consider a positively ground-preserving variant of Example 11:

$$\mathsf{P}(0) \qquad (i) \qquad \neg\mathsf{P}(x) \vee \mathsf{P}(\mathsf{f}(\mathsf{f}(x))) \qquad (ii) \qquad \neg\mathsf{P}(x) \vee \neg\mathsf{P}(\mathsf{f}(x)) \qquad (iii)$$

The finite model property implies that it is essentially finite. SGGS terminates with neither I^- nor I^+ as initial interpretation. With I^- it generates:

$$\begin{array}{lll}
\varepsilon & \vdash [\mathsf{P}(0)] & \mathsf{extend}\ (i) \\
& \vdash [\mathsf{P}(0)],\ \neg\mathsf{P}(0) \vee [\mathsf{P}(\mathsf{f}^2(0))] & \mathsf{extend}\ (ii) \\
& \vdash [\mathsf{P}(0)],\ \neg\mathsf{P}(0) \vee [\mathsf{P}(\mathsf{f}^2(0))],\ \neg\mathsf{P}(\mathsf{f}^2(0)) \vee [\mathsf{P}(\mathsf{f}^4(0))] & \mathsf{extend}\ (ii) \\
& \vdash \ldots &
\end{array}$$

Positive hyperresolution generates the series $\{\mathsf{P}(\mathsf{f}^{2k}(0))\}_{k \geq 0}$, which is essentially the same behavior as SGGS. DPLL($\Gamma + \mathcal{T}$) tries $\mathsf{f}(x) \simeq x$, detects a conflict, backtracks, tries $\mathsf{f}^2(x) \simeq x$, and halts reporting satisfiability.

Ensuring termination by restricting new (i.e., non-input) terms to come from a finite basis is common for conflict-driven decision procedures [11,16]. A major direction for future work towards decision procedures for richer languages is the integration of SGGS with CDSAT [11], in order to endow SGGS with equality and CDSAT with quantifiers. Speculative inferences and initial interpretations not based on sign are additional leads. For the restrained fragment, one may consider its relations with other decidable fragments and its relevance to applications beyond TPTP. While positive clauses in a restrained set are ground, one may study the decidability of sets where positive clauses are not necessarily ground, but admit a *restraining* rewrite system (Definition 7) such that *narrowing* halts. Techniques to detect the termination of narrowing are known [35].

Although Koala is only a prototype, the experiments show potential and allow us to identify critical issues for the performance of an SGGS prover. For example, instance generation by SGGS-extension is a bottleneck for problems with many input clauses, and forms of *caching* should be considered to avoid repeating computations. Further development of Koala and more experiments may contribute to discover classes of first-order problems where the conflict-driven style of SGGS reasoning is especially rewarding.

References

1. Abadi, A., Rabinovich, A., Sagiv, M.: Decidable fragments of many-sorted logic. J. Symb. Comput. **45**(2), 153–172 (2010)
2. Ackermann, W.: Solvable Cases of the Decision Problem. North Holland, Amsterdam (1954)
3. Alagi, G., Weidenbach, C.: NRCL - a model building approach to the Bernays-Schönfinkel fragment. In: Lutz, C., Ranise, S. (eds.) FroCoS 2015. LNCS (LNAI), vol. 9322, pp. 69–84. Springer, Cham (2015). https://doi.org/10.1007/978-3-319-24246-0_5
4. Andréka, H., van Benthem, J., Nemeti, I.: Modal logics and bounded fragments of predicate logic. J. Phil. Logic **27**(3), 217–274 (1998)
5. Bachmair, L., Ganzinger, H., Waldmann, U.: Superposition with simplification as a decision procedure for the monadic class with equality. In: Gottlob, G., Leitsch, A., Mundici, D. (eds.) KGC 1993. LNCS, vol. 713, pp. 83–96. Springer, Heidelberg (1993). https://doi.org/10.1007/BFb0022557
6. Baumgartner, P., Fuchs, A., Tinelli, C.: Implementing the model evolution calculus. Int. J. Artif. Intell. Tools **15**(1), 21–52 (2006)
7. Baumgartner, P., Tinelli, C.: The model evolution calculus as a first-order DPLL method. Artif. Intell. **172**(4–5), 591–632 (2008)
8. Bernays, P., Schönfinkel, M.: Zum Entscheidungsproblem der mathematischen Logik. Math. Annalen **99**, 342–372 (1928)
9. Blackburn, P., de Rijke, M., Venema, Y.: Modal Logic. Cambridge University Press, Cambridge (2001)
10. Bonacina, M.P., Dershowitz, N.: Canonical ground Horn theories. In: Voronkov, A., Weidenbach, C. (eds.) Programming Logics. LNCS, vol. 7797, pp. 35–71. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37651-1_3
11. Bonacina, M.P., Graham-Lengrand, S., Shankar, N.: Conflict-driven satisfiability for theory combination: transition system and completeness. J. Autom. Reason. **64**(3), 579–609 (2020)
12. Bonacina, M.P., Lynch, C.A., de Moura, L.: On deciding satisfiability by theorem proving with speculative inferences. J. Autom. Reason. **47**(2), 161–189 (2011)
13. Bonacina, M.P., Plaisted, D.A.: Semantically-guided goal-sensitive reasoning: model representation. J. Autom. Reason. **56**(2), 113–141 (2016)
14. Bonacina, M.P., Plaisted, D.A.: Semantically-guided goal-sensitive reasoning: inference system and completeness. J. Autom. Reason. **59**(2), 165–218 (2017)
15. Caferra, R., Leitsch, A., Peltier, N.: Automated Model Building. Kluwer, Alphen aan den Rijn (2004)

16. de Moura, L., Jovanović, D.: A model-constructing satisfiability calculus. In: Giacobazzi, R., Berdine, J., Mastroeni, I. (eds.) VMCAI 2013. LNCS, vol. 7737, pp. 1–12. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-35873-9_1
17. Fermüller, C., Leitsch, A.: Model building by resolution. In: Börger, E., Jäger, G., Kleine Büning, H., Martini, S., Richter, M.M. (eds.) CSL 1992. LNCS, vol. 702, pp. 134–148. Springer, Heidelberg (1993). https://doi.org/10.1007/3-540-56992-8_10
18. Fermüller, C., Leitsch, A., Hustadt, U., Tammet, T.: Resolution decision procedures. In: Handbook of Automated Reasoning, pp. 1791–1849. Elsevier and MIT Press (2001)
19. Fiori, A., Weidenbach, C.: SCL clause learning from simple models. In: Fontaine, P. (ed.) CADE 2019. LNCS (LNAI), vol. 11716, pp. 233-249. Springer, Cham (2019). https://doi.org/10.1007/978-3-030-29436-6_14
20. Ganzinger, H., de Nivelle, H.: A superposition decision procedure for the guarded fragment with equality. In: Proceedings of LICS-14. IEEE Computer Society Press (1999)
21. Grädel, E., Kolaitis, P., Vardi, M.: On the decision problem for two-variable first-order logic. Bull. Symb. Log. **3**, 53–69 (1997)
22. Hustadt, U., Schmidt, R.A., Georgieva, L.: A survey of decidable first-order fragments and description logics. J. Relat. Methods Comput. Sci. **1**, 251–276 (2004)
23. Joyner Jr., W.H.: Resolution strategies as decision procedures. J. ACM **23**(3), 398–417 (1976)
24. Korovin, K.: Inst-Gen – a modular approach to instantiation-based automated reasoning. In: Voronkov, A., Weidenbach, C. (eds.) Programming Logics. LNCS, vol. 7797, pp. 239–270. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-37651-1_10
25. Korovin, K.: Non-cyclic sorts for first-order satisfiability. In: Fontaine, P., Ringeissen, C., Schmidt, R.A. (eds.) FroCoS 2013. LNCS (LNAI), vol. 8152, pp. 214–228. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-40885-4_15
26. Korovin, K.: iProver – an instantiation-based theorem prover for first-order logic (v2.8) (2018). http://www.cs.man.ac.uk/korovink/iprover/index.html
27. Korp, M., Sternagel, C., Zankl, H., Middeldorp, A.: Tyrolean Termination Tool 2. In: Treinen, R. (ed.) RTA 2009. LNCS, vol. 5595, pp. 295–304. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-642-02348-4_21
28. Kounalis, E., Rusinowitch, M.: On word problems in Horn theories. J. Symb. Comput. **11**(1–2), 113–128 (1991)
29. Lahiri, S., Qadeer, S.: Back to the future: revisiting precise program verification using SMT solvers. In: Proceedings of POPL-31, pp. 171–182. ACM (2004)
30. Lamotte-Schubert, M., Weidenbach, C.: BDI: a new decidable clause class. J. Log. Comput. **27**(2), 441–468 (2017)
31. Lewis, C.I., Langford, C.H.: Symbolic Logic. Dover Publications Inc., Mineola (1932)
32. Lutz, C., Sattler, U., Wolter, F.: Modal logic and the two-variable fragment. In: Fribourg, L. (ed.) CSL 2001. LNCS, vol. 2142, pp. 247–261. Springer, Heidelberg (2001). https://doi.org/10.1007/3-540-44802-0_18
33. McMillan, K.: Developing distributed protocols with Ivy. Slides (2019). http://vmcaischool19.tecnico.ulisboa.pt/
34. Navarro, J.A., Voronkov, A.: Proof systems for effectively propositional logic. In: Armando, A., Baumgartner, P., Dowek, G. (eds.) IJCAR 2008. LNCS (LNAI), vol. 5195, pp. 426–440. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-71070-7_36

35. Nishida, N., Vidal, G.: Termination of narrowing via termination of rewriting. Appl. Algebr. Eng. Commun. **21**(3), 177–225 (2010)
36. Padon, O., McMillan, K., Panda, A., Sagiv, M., Shoham, S.: Ivy: Safety verification by interactive generalization. In: SIGPLAN Not. vol. 51, no. 6, pp. 614–630 (2016)
37. Piskac, R., de Moura, L., Bjørner, N.: Deciding effectively propositional logic using DPLL and substitution sets. J. Autom. Reason. **44**(4), 401–424 (2010)
38. Plaisted, D.A., Zhu, Y.: The Efficiency of Theorem Proving Strategies. Friedr. Vieweg, Hamburg (1997)
39. Ramsey, F.: On a problem in formal logic. Proc. Lond. Math. Soc. **30**, 264–286 (1930)
40. Rubio, A.: A fully syntactic AC-RPO. Inform. Comput. **178**(2), 515–533 (2002)
41. Sutcliffe, G.: The TPTP problem library and associated infrastructure from CNF to TH0, TPTP v6.4.0. J. Autom. Reason. **59**(4), 483–502 (2017)