SGGS: A CDCL-like first-order theorem-proving method[1]

Maria Paola Bonacina

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

Formal Topics Series
Computer Science Laboratory, SRI International
Menlo Park, California, USA

December 2015 and January 2016

[1] Joint work with David A. Plaisted

Logical methods for machine intelligence

- Theorem provers for higher order logic (ITP, HOL)
- Theorem provers for first order logic (ATP, FOL)
- Solvers for satisfiability modulo theories (SMT)
- Solvers for satisfiability in propositional logic (SAT)
- ....

Solvable / Unsolvable

- Solver: decidable problem
  - SAT
  - SMT
- Prover: undecidable problem
  - ATP/FOL: validity semi-decidable, satisfiability not
  - ITP/HOL: neither

Proof theory / Model theory

- ITP/HOL
  - Direct proof construction
  - Foundation: proof theory
- ATP/FOL, SMT, SAT/PL
  - Proofs by refutation
  - Inconsistency reveals unsatisfiability: no model
  - Search for model
  - Foundation: model theory
    - ATP/FOL: proof by refutation
    - SMT, SAT/PL: either

Models

- SAT/PL
  - n propositional variables
  - 2^n interpretations
  - Survey: semantic tree
- ATP/FOL
  - Clausal form
  - Herbrand interpretations: Herbrand universe, Herbrand base
  - Powerset of the Herbrand base
  - Survey: infinite semantic tree

How to reason with and about first-order models?

Semantic resolution

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

Hyperresolution

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

Semantic guidance

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

Examples: semantic resolution, hyperresolution

Resolution with set of support

- $H \models^? \varphi$
- $H \cup \{\neg\varphi\} \vdash^? \bot$
- $H \cup \{\neg\varphi\} \rightsquigarrow S$ set of clauses to be refuted
- $S = T \uplus SOS$ where $\{\neg\varphi\} \rightsquigarrow SOS$ and $T = S \setminus SOS$ is consistent: $\mathcal{I} \models T$
- Allow resolution only if at least a parent is from SOS
- Add all resolvents to SOS
- Instance of semantic resolution where $\mathcal{I} \models T$ and $\mathcal{I} \not\models SOS$

Goal sensitivity

A reasoning method is goal sensitive if it generates only clauses connected with the negation of the conjecture (the goal).

May be relevant in case of large axiom sets or knowledge bases.

Example: resolution with set of support

DPLL

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

Clausal propagation

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

DPLL-CDCL or CDCL tout court

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

Model-based reasoning

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

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

Examples: DPLL, CDCL

Proof confluence

- Resolution vs. tableaux debate
- Confluence: diamond property: $\swarrow \searrow \Rightarrow \searrow \swarrow$
- Proof confluence:
  Committing to an inference never prevents proof
- No backtracking
- Resolution is proof confluent, tableaux are not
- Backtracking in DPLL and CDCL: from a branch to another
- Backtracking in tableaux: from a tableau to another (rigid variables)

The quest

A theorem-proving method simultaneously

- First order
- Semantically guided
- Goal sensitive
- Model based
- Proof confluent

SGGS: Semantically-Guided Goal Sensitive reasoning

A new method for first-order theorem proving that is

- Semantically guided
- Goal sensitive (with flexibility)
- Model based
- Proof confluent

and that

- Lifts CDCL to first-order logic

SGGS basics

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

Semantic guidance for model-based reasoning I

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

Semantic guidance for model-based reasoning II

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

SGGS clause sequence

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

Examples

- $\mathcal{I}$: all negative
- A sequence of unit clauses:
  $[P(a,x)], [P(b,y)], [\neg P(z,z)], [P(u,v)]$
- A sequence of non-unit clauses:
  $[P(x)],\ \neg P(f(y)) \vee [Q(y)],\ \neg P(f(z)) \vee \neg Q(g(z)) \vee [R(f(z),g(z))]$
- A sequence of constrained clauses:
  $[P(x)],\ top(y) \neq g \rhd [Q(y)],\ z \not\equiv c \rhd [Q(g(z))]$

Candidate partial model represented by Γ

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

Candidate partial model represented by Γ

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

  $L_i\sigma$ is a proper ground instance

Example

- Sequence Γ: $[P(a,x)], [P(b,y)], [\neg P(z,z)], [P(u,v)]$
- Partial model $\mathcal{I}^p(\Gamma)$:
  $\mathcal{I}^p(\Gamma) \models P(a,t)$ for all ground terms t
  $\mathcal{I}^p(\Gamma) \models P(b,t)$ for all ground terms t
  $\mathcal{I}^p(\Gamma) \models \neg P(t,t)$ for t other than a and b
  $\mathcal{I}^p(\Gamma) \models P(s,t)$ for all distinct ground terms s and t

Model represented by Γ

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

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

Example

- $\mathcal{I}$: all negative
- Sequence Γ: $[P(a,x)], [P(b,y)], [\neg P(z,z)], [P(u,v)]$
- Represented model $\mathcal{I}[\Gamma]$:
  $\mathcal{I}[\Gamma] \models P(a,t)$ for all ground terms t
  $\mathcal{I}[\Gamma] \models P(b,t)$ for all ground terms t
  $\mathcal{I}[\Gamma] \models \neg P(t,t)$ for t other than a and b
  $\mathcal{I}[\Gamma] \models P(s,t)$ for all distinct ground terms s and t
  $\mathcal{I}[\Gamma] \not\models L$ for all other positive literals L

Disjoint prefix

The disjoint prefix of Γ is

- The longest prefix of Γ where every selected literal contributes to $\mathcal{I}[\Gamma]$ all its ground instances
- That is, where all ground instances are proper
- Intuitively, a polished portion of Γ

First-order clausal propagation

- Consider a literal M selected in clause C_j in Γ, and a literal L in C_i, $i > j$:
  $\ldots, \ldots \vee [M] \vee \ldots, \ldots, \ldots \vee L \vee \ldots, \ldots$
  If all ground instances of L appear negated among the proper ground instances of M, L is uniformly false in $\mathcal{I}[\Gamma]$
- L depends on M, like $\neg L$ depends on L in propositional clausal propagation when L is in the trail
- Since every literal in Γ is either $\mathcal{I}$-true or $\mathcal{I}$-false, M will be one and L the other

Example

- $\mathcal{I}$: all negative
- Sequence Γ:
  $[P(x)]$, $\neg P(f(y)) \vee [Q(y)]$, $\neg P(f(z)) \vee \neg Q(g(z)) \vee [R(f(z), g(z))]$
- $\neg P(f(y))$ is made uniformly false in $\mathcal{I}[\Gamma]$ by $[P(x)]$
- $\neg P(f(z))$ is made uniformly false in $\mathcal{I}[\Gamma]$ by $[P(x)]$
- $\neg Q(g(z))$ is made uniformly false in $\mathcal{I}[\Gamma]$ by $[Q(y)]$

First-order clausal propagation

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

Semantically-guided first-order clausal propagation

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

How does SGGS build clause sequences?

- Main inference rule: SGGS-extension
- $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$
- $\mathcal{I}[\Gamma] \not\models C'$ for some ground instance C' of C
- Then SGGS-extension uses Γ and C to generate a (possibly constrained) clause $A \triangleright E$ such that
  - E is an instance of C
  - C' is a ground instance of $A \triangleright E$

  and adds it to Γ to get Γ'

How can a ground clause be false I

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

For each literal L of C':

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

The SGGS-extension inference scheme I

- Unify literals $L_1, \ldots, L_n$ $(n \geq 1)$ of C with $\mathcal{I}$-false selected literals $M_1, \ldots, M_n$ of opposite sign in Γ: most general unifier α
- Generate instance $C\alpha$

- The $L_1\alpha, \ldots, L_n\alpha$ are $\mathcal{I}$-true
- The $M_1, \ldots, M_n$ are those that make the $\mathcal{I}$-true literals of C' false in $\mathcal{I}[\Gamma]$
- Instance generation is guided by the current model $\mathcal{I}[\Gamma]$

The SGGS-extension inference scheme II

- ϑ semantic falsifier for C: all literals in $C\vartheta$ are $\mathcal{I}$-false
- Most general semantic falsifier

- β most general semantic falsifier of $(C \setminus \{L_1, \ldots, L_n\})\alpha$
- Generate instance $C\alpha\beta$ where the $L_1\alpha\beta, \ldots, L_n\alpha\beta$ are $\mathcal{I}$-true and all other literals are $\mathcal{I}$-false

Non-empty for non-trivial $\mathcal{I}$

Example

- S contains $\{P(a), \neg P(x) \vee Q(f(y)), \neg P(x) \vee \neg Q(z)\}$
- $\mathcal{I}$: all negative
- Γ_0 is empty
  $\mathcal{I}[\Gamma_0] = \mathcal{I} \not\models P(a)$
- $\Gamma_1 = [P(a)]$ with α and β empty
- $\mathcal{I}[\Gamma_1] \not\models \neg P(x) \vee Q(f(y))$
- $\Gamma_2 = [P(a)],\ \neg P(a) \vee [Q(f(y))]$
  with $\alpha = \{x \leftarrow a\}$ and β empty

How can a ground clause be false II

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

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

Three kinds of SGGS-extension

The added clause E is

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

Lifting theorem for SGGS-extension

If $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$
($\mathcal{I}[\Gamma] \not\models C'$ for C' ground instance of C)
then there is a (possibly constrained) clause $A \rhd E$ such that

- E is an instance of C
- C' is a ground instance of $A \rhd E$
- $A \rhd E$ can be added to Γ by SGGS-extension to get Γ'

Example (continued)

- S contains $\{P(a), \neg P(x) \vee Q(f(y)), \neg P(x) \vee \neg Q(z)\}$
- $\mathcal{I}$: all negative
- After two non-conflicting SGGS-extensions:
  $\Gamma_2 = [P(a)],\ \neg P(a) \vee [Q(f(y))]$
- $\mathcal{I}[\Gamma_2] \not\models \neg P(x) \vee \neg Q(z)$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(w))]$ with $\alpha = \{x \leftarrow a, z \leftarrow f(y)\}$ plus renaming
- Conflict! with $\mathcal{I}$-all-true conflict clause

Conflict handling in SGGS

The conflict clause is

- $\mathcal{I}$-all-true: solve the conflict
- Non-$\mathcal{I}$-all-true: explain and solve the conflict

First-order conflict explanation: SGGS-resolution

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

First-order conflict-solving: SGGS-move

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

Example (continued)

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

Further elements

- There's more to SGGS: first-order literals may intersect having ground instances with the same atom
- SGGS uses splitting inference rules to partition clauses and isolate intersections that can then be removed by SGGS-resolution (different sign) or SGGS-deletion (same sign)
- Splitting introduces constraints that are a kind of Herbrand constraints (e.g., $x \not\equiv y \triangleright P(x, y)$, $top(y) \neq g \triangleright Q(y)$)
- SGGS works with constrained clauses

Theorems

SGGS is

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

Bundled derivation

Bundled derivation: all conflicting SGGS-extension followed by explanation by SGGS-resolution and conflict solving by SGGS-move

Refutational completeness

- S: input set of clauses
- S unsatisfiable: any fair SGGS-derivation terminates with refutation
- S satisfiable: derivation may be infinite; its limiting sequence represents a model

Proof of refutational completeness: building blocks

- A convergence ordering $>^c$ on clause sequences: ensures that there is no infinite descending chain of sequences of bounded length
- A notion of fairness for SGGS-derivations: ensures that the procedure does not ignore inferences on shorter prefixes to work on longer ones
- A notion of limiting sequence for SGGS-derivations: every prefix stabilizes eventually

Convergence ordering I

- Quasi-orderings $\geq_i$ and equivalence relations $\approx_i$ on clause sequences of length up to i
- Convergence ordering $>^c$: lexicographic combination of $>_i$'s
- Equivalence relation $\approx^c$: same length and all prefixes in the $\approx_i$'s

Convergence ordering II

Theorem:
$>_i$ is well-founded on clause sequences of length at least i

Corollary:
Descending chain $\Gamma^1 >^c \Gamma^2 >^c \ldots \Gamma^j >^c \Gamma^{j+1} >^c \ldots$
of sequences of bounded length (for all j, $|\Gamma^j| \leq n$) is finite

No infinite descending chain of sequences of bounded length

Fairness I

- Index of inference $\Gamma \vdash \Gamma'$:
  the shortest prefix that gets reduced
  the smallest i such that $\Gamma|_i >^c \Gamma'|_i$
- Index(Γ): minimum index of any inference applicable to Γ

Fairness II

Fair derivation $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_j \vdash \ldots$:
$\forall i,\ i > 0$, if for infinitely many Γ_j's *index*$(\Gamma_j) \leq i$
for infinitely many Γ_j's the applied inference has index $\leq i$

Any SGGS-inference that is infinitely often possible is eventually done

Example: the minimal index SGGS-strategy that always selects an inference of minimal index is trivially fair

Limiting sequence

- Derivation $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \vdash \Gamma_j \vdash \ldots$ admits limit
  if there exists a Γ (limit)
  such that for all lengths i, $i \leq |\Gamma|$
  there is an integer n_i
  such that for all indices $j \geq n_i$ in the derivation
  if $|\Gamma_j| \geq i$ then $\Gamma_j|_i \approx^c \Gamma|_i$
- Every prefix stabilizes eventually
- The longest such sequence Γ_∞ is the limiting sequence
- Both derivation and Γ_∞ may be finite or infinite

Convergence and descending chain theorems

- Convergence theorem:
  A derivation that is a non-ascending chain admits limiting sequence
- Descending chain theorem:
  A bundled derivation forms a descending chain

Completeness theorem

Theorem:
For all initial interpretations $\mathcal{I}$ and sets S of first-order clauses, if S is unsatisfiable, any fair bundled SGGS-derivation is a refutation

Idea of proof:
If not, infinitely many SGGS-extensions apply;
infinite derivation with infinite limiting sequence Γ_∞;
Γ_j gets reduced in $>^c$ in a finite prefix $(\Gamma_j)|_n$ that had already converged $((\Gamma_j)|_n = (\Gamma_\infty)|_n)$: contradiction

Summary

SGGS is possibly unique in being simultaneously

- First order
- Model based à la CDCL
- Semantically guided
- Refutationally complete
- Goal sensitive (when deemed desirable)
- Proof confluent

References on SGGS

- Semantically-guided goal-sensitive reasoning: model representation. *Journal of Automated Reasoning*, 29 pages, published online June 26, 2015.
- Semantically-guided goal-sensitive reasoning: inference system and completeness. Submitted November 9, 2015, 56 pages.
- SGGS theorem proving: an exposition. 4th Workshop on Practical Aspects in Automated Reasoning (PAAR), Vienna, July 2014. EPiC 31:25-38, July 2015.
- Constraint manipulation in SGGS. 28th Workshop on Unification (UNIF), Vienna, July 2014. TR 14-06, RISC, 47–54, 2014.

Future work on SGGS

- Implementation: algorithms and strategies
- Non-trivial initial interpretations?
- Extension to equality?
- SGGS for model building?
- SGGS for decision procedures for decidable fragments?

Towards a semantically-oriented style of theorem proving that may pay off for hard problems or new domains

Future work in general

- ITP/HOL: Instance generation for PVS?
- SMT: Boolean Algebra with Presburger Arithmetic: Boolean ring?
- ATP/FOL: SGGS