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

Maria Paola Bonacina

Dipartimento di Informatica
Università degli Studi di Verona
Verona, Italy

April 2016

[1] Joint work with David Plaisted

Big picture: Model-based reasoning

- Derivation state includes candidate partial model
- Inference and search for model guide each other
- Inference as model transformation
- E.g., SAT-solving, SMT-solving, MBTC, MCsat, Model-based projections ....

Example: the CDCL procedure for PL

- Propositional logic
- A set of clauses S to be satisfied or refuted
- Model representation: trail of literals
- Search for model: decision, backjumping
- Inference: clausal propagation, explanation, learning

Semantics in first-order reasoning

- Semantic resolution
- Hyperresolution
- Resolution with set of support
- Model elimination and tableaux-based methods: an open branch is a candidate model
- Instance-based methods where instance generation is model driven
- ....

SGGS: Semantically-Guided Goal-Sensitive reasoning

- Model-based: It lifts CDCL to first-order logic
- Also semantically guided, goal sensitive, proof confluent
- Refutationally complete: S unsatisfiable, SGGS gets $\perp$
- Explicit model construction: S is satisfiable, the limit of the derivation is a model

Model representation in PL

- Propositional logic
- Propositional variable P is either true or false
- 2^n interpretations for n propositional variables
- Guess P (or $\neg P$)

Model representation in FOL

- First-order logic
- Clausal form, Herbrand interpretations
- $P(x)$ has infinitely many ground instances: $P(a)$, $P(f(a))$, $P(f(f(a)))$ ... infinite Herbrand base
- Infinitely many interpretations where each ground instance is either true or false: powerset of the Herbrand base
- What do we guess?! How do we get started?!
- Answer: Semantic guidance

Semantic guidance

- Take $\mathcal{I}$ with all positive ground literals true
- S: set of clauses to be satisfied or refuted
- $\mathcal{I} \models S$: done! $\mathcal{I} \not\models S$: modify $\mathcal{I}$ to satisfy S
- How? Flipping literals from positive to negative
- SGGS discovers which negative literals are needed
- Initial interpretation $\mathcal{I}$: starting point in the search for a model and default interpretation

SGGS basics

- Set S of clauses to refute or satisfy
- Initial fixed Herbrand interpretation $\mathcal{I}$, e.g.:
  - All negative (as in positive hyperresolution)
  - All positive (as in negative hyperresolution)
  - $\mathcal{I} \not\models SOS$, $\mathcal{I} \models T$ (as in resolution with set of support)
  - Other (e.g., $\mathcal{I}$ satisfies the axioms of a theory $\mathcal{T}$ and we have a model constructing $\mathcal{T}$-solver acting as oracle)
- $\mathcal{I} \models S$: problem solved
- Otherwise: modify $\mathcal{I}$ to satisfy S
- How to represent this modified interpretation?

Uniform falsity

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

Truth and uniform falsity in the initial interpretation

- $\mathcal{I}$-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 (invariant)
- 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 (to change $\mathcal{I}$)
- 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
- Literal selection in SGGS corresponds to decision in CDCL

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 try 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 $dp(\Gamma)$ 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
- No two selected literals in the disjoint prefix intersect
- Intuitively, a polished portion of Γ

Examples

$[P(a,x)], [P(b,y)], [\neg P(z,z)], [P(u,v)]$:
the disjoint prefix is $[P(a,x)], [P(b,y)]$

$[P(x)],\ \neg P(f(y)) \vee [Q(y)],\ \neg P(f(z)) \vee \neg Q(g(z)) \vee [R(f(z), g(z))]$:
the disjoint prefix is the whole sequence

$[P(x)],\ top(y) \neq g \rhd [Q(y)],\ z \not\equiv c \rhd [Q(g(z))]$:
the disjoint prefix is the whole sequence

Propositional 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

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:
  $[P(x)],\ \neg P(f(y)) \vee [Q(y)],\ \neg P(f(z)) \vee \neg Q(g(z)) \vee [R(f(z), g(z))]$
- $\neg P(f(y))$ depends on $[P(x)]$
- $\neg P(f(z))$ depends on $[P(x)]$
- $\neg Q(g(z))$ depends on $[Q(y)]$

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 assignments to keep track of the dependences of $\mathcal{I}$-true literals on selected $\mathcal{I}$-false literals
- Non-selected $\mathcal{I}$-true literals are assigned (invariant)
- 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

SGGS-derivation

- Set S of clauses to refute or satisfy
- Initial interpretation $\mathcal{I}$
- $(S; \mathcal{I}; \Gamma_0) \vdash (S; \mathcal{I}; \Gamma_1) \vdash \ldots (S; \mathcal{I}; \Gamma_i) \vdash (S; \mathcal{I}; \Gamma_{i+1}) \vdash \ldots$
- $\Gamma_0 \vdash \Gamma_1 \vdash \ldots \Gamma_i \vdash \Gamma_{i+1} \vdash \ldots$

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 $A \rhd E$ such that
  - E is an instance of C
  - C' is a ground instance of $A \rhd E$

  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

- Clause $C \in S$: main premise
- 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 $dp(\Gamma)$: most general unifier α
- Clauses where the $M_1, \ldots, M_n$ are selected: side premises
- Generate instance $C\alpha$

The SGGS-extension inference scheme

- 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 $\mathcal{I}[\Gamma]$
- The $M_1, \ldots, M_n$ are $\mathcal{I}$-false but true in $\mathcal{I}[\Gamma]$: instance generation is guided by the current model $\mathcal{I}[\Gamma]$

Semantic falsifier

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

The SGGS-extension inference scheme

- 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
- Assign the $\mathcal{I}$-true literals of $C\alpha\beta$ to the side premises
- $C\alpha\beta$ is called extension clause

β not-empty only for $\mathcal{I}$ not based on sign

Examples

- 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: all its literals are assigned and depend on selected $\mathcal{I}$-false literals in Γ;
  C' is instance of an $\mathcal{I}$-all-true conflict clause
- Or C' has $\mathcal{I}$-false literals and all of them depend on selected $\mathcal{I}$-true literals in Γ;
  C' is instance of a non-$\mathcal{I}$-all-true conflict clause
- Or C' has $\mathcal{I}$-false literals and at least one of them is not interpreted by $\mathcal{I}^p(\Gamma)$: C' is a proper ground instance of some clause

Three kinds of SGGS-extension

The extension clause 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

SGGS-extension with $\mathcal{I}$-all-true conflict clause

The extension clause $E = C\alpha\beta$ is an $\mathcal{I}$-all-true conflict clause:

$$\frac{\Gamma}{\Gamma A \rhd E[L]}$$

- Constraints may be inherited from the side premises
- L is the literal assigned to the side premise of largest index: the selected literal in an $\mathcal{I}$-all-true conflict clause is assigned rightmost

SGGS-extension with non-$\mathcal{I}$-all-true conflict clause

All I-false literals in the extension clause $E = C\alpha\beta$ intersect $\mathcal{I}$-true selected literals in $dp(\Gamma)$:

$$\frac{\Gamma}{\Gamma A\lambda \triangleright E[L]\lambda}$$

- Extension substitution λ: most general substitution that let the I-false literals in the extension clause depend on I-true selected literals in $dp(\Gamma)$; in practice: most general unifier
- L is an arbitrary $\mathcal{I}$-false literal: heuristic choice

Non-conflicting SGGS-extension

The extension clause $E = C\alpha\beta$ has $\mathcal{I}$-false literals with proper ground instances w.r.t. Γ:

$$\frac{\Gamma}{\Gamma A \triangleright E[L]}$$

- L is an arbitrary $\mathcal{I}$-false literal with proper ground instances: heuristic choice

Non-conflicting SGGS-extension

The extension clause $E = C\alpha\beta$ has an I-false literal L with proper ground instances w.r.t. a prefix Γ^1 of Γ:

$$\frac{\Gamma^1 J \rhd N[O]\Gamma^2}{\Gamma^1 A \rhd E[L]\Gamma^2}$$

- $A \rhd E[L]$ has smaller proper ground instances than $J \rhd N[O]$ in a total well-founded ordering on ground literals that extends the size ordering
- All side premises are in Γ^1
- Γ^1 is the shortest such prefix

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

- 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

CDCL

- 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

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

First-order conflict explanation: SGGS-resolution

- 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 (thanks to extension substitution) 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 does not bother other assignments because L was assigned rightmost
- 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 $\{P(a), \neg P(x) \vee Q(f(y)), \neg P(x) \vee \neg Q(z)\}$
- $\Gamma_3 = [P(a)],\ \neg P(a) \vee [Q(f(y))],\ \neg P(a) \vee [\neg Q(f(w))]$
- $\Gamma_4 = [P(a)],\ \neg P(a) \vee [\neg Q(f(w))],\ \neg P(a) \vee [Q(f(y))]$
- $\Gamma_5 = [P(a)],\ \neg P(a) \vee [\neg Q(f(w))],\ [\neg P(a)]$
- $\Gamma_6 = [\neg P(a)],\ [P(a)],\ \neg P(a) \vee [\neg Q(f(w))]$
- $\Gamma_7 = [\neg P(a)],\ \square,\ \neg P(a) \vee [\neg Q(f(w))]$
- Refutation!

Bundled derivation

All conflicting SGGS-extension are followed by (bundled with) explanation by SGGS-resolution and conflict solving by SGGS-move

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., $x \not\equiv y \rhd P(x, y)$, $top(y) \neq g \rhd Q(y)$)
- SGGS works with constrained clauses

SGGS makes progress

For all states $(S, I; \Gamma)$:

- If $\mathcal{I}[\Gamma] \not\models C$ for some clause $C \in S$ and $\Gamma = dp(\Gamma)$, SGGS-extension applies to Γ
- If $\Gamma \neq dp(\Gamma)$, an SGGS inference rule other than SGGS-extension applies to Γ

Refutational completeness and goal-sensitivity

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 $S = T \uplus SOS$

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 $\Gamma^1 >^c \Gamma^2 >^c \dots \Gamma^j >^c \Gamma^{j+1} >^c \dots$
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 $\mathit{index}(\Gamma_j) \leq i$
for infinitely many Γ_j's the applied inference has index $\leq i$

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

The minimal index SGGS-strategy that always selects an inference of minimal index is 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* 56(2):113–141, February 2016.
- Semantically-guided goal-sensitive reasoning: inference system and completeness. Submitted, 58 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
- Heuristic choices: literal selection, assignments
- Simpler SGGS?
- Initial interpretations not based on sign
- 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