A Superposition Decision Procedure for the Guarded Fragment with Equality

Authors:
Harald Ganzinger
Hans de Nivelle

Presented by:
Syed Fawad Mustafa

Introduction

- Spanish saying:

  "Good things in life come in small packages."
- Our interest is in "Logic and Computation" being decidable but FO(First Order logic) is semi−decidable.
- Instead we may use fragments of FO where decidability holds.

Why GF?

- Use of quantifiers in restricted way in FO hold good properties of modal fragments.
  - (e.g: In ML ◊ and ❑ can be translated to ∀ and ∃ in FO)
- If ML are translated to FO, FO formulae in GF are obtained.
- In GF only monadic or binary predicates are used
- GF imposes all the quantifiers to be guarded by atomic formulae.

Reference:

- "Guarded Fragments of First Order logic: Perspective for new Description logic" by Erich Gradel.
- "The Guarded Fragment: Ins and Outs" by Carlos Areces, Christof Monz, Hans de Nivelle and Maarten de Rijke.

Outline

- The decision procedure for GF is based on resolution with SP.
- This method will be more useful in practice then methods based on the enumeration of certain finite structures.
- No need of any sophisticated simplification and redundancy elimination method to make SP terminate on the class of clauses obtained from clausification of guarded formulas.
- Optimal with regard to time complexity.
- Re−usable results about refutational completeness.

The Guarded Fragment

- Following are the inductively defined formulae of GF of function−free first order logic:
  - $\top$ and $\bot$ are in GF.
  - If A is an atom then A is in GF.
  - GF is closed under boolean combinations.
  - If F $\in$ GF and G is an atom, for which every free variable of F is among the arguments of G, then:
    - $\forall\bar{x}\ (G \rightarrow F) \in GF$ (or, equivalently, $\forall\bar{x}(\neg G \vee F) \in GF$) and $\exists\bar{x}(G \wedge F) \in GF$, for every sequence $\bar{x}$ of variables.

contd...

- The atoms G which appear as constraints for quantified variables are called guards. Following are the examples of guards formulae:
  - $\forall x\ (x \approx x \rightarrow p(x))$, $\exists x\ (p(x) \wedge q(x))$
  - $\forall yz\ (r(y,y,z) \rightarrow \bot)$, $\forall xy (r(x,y) \rightarrow r(y,x))$
  - $\forall xy\ (r(x,y) \rightarrow \exists z\ r(y,z))$

contd...

- $\exists x\, [R(w,x) \wedge \forall y(R(x,y) \rightarrow p(y)) \wedge q(x)]$
  - This formula is the translation of the modal formula $\Diamond(\Box p \wedge q)$ with respect to a world w.

Non−guarded Formulae

- $\forall xy\ p(x,y)$
- $\forall x_1 x_2 x_3\ [p(x_1, x_2) \rightarrow p(x_2, x_2) \rightarrow p(x_1,x_3)]$

* The last formula states the transitivity of p. As this is not guarded, for modal logics such as S4 which are based on transitive frames the standard embedding methods lead out−side the guarded fragment.

The Superposition Calculus

- **Equality(≈):**
  - Do not distinguish b/w equations s≈t and t≈s.
- **The Calculus is clausal i.e**
  - Clauses are multisets of literals $L_1, \ldots, L_k$, $k \geq 0$ OR

    $L_1 \vee \ldots \vee L_k$(Disjunction).
- Calculus can be parameterized by admissible orderings and selection function for negative literals, for each setting of two parameters it is ***refutationally complete.***

Contd...

- Ordering with non−equational atoms
  - $p(t_1,...,t_k)$
- Admissible Ordering (i.e: >)
  - Example: $\underline{p(f(x))} \wedge p(x)$
- Selection Function (i.e: Σ)
  - Example: $p(f(x)) \wedge \underline{\neg p(x)}$
- *Eligible* literals

Inference Rules

- **Ordered Factoring**
  - $$\frac{A_1 \vee A_2 \vee R}{A_1\sigma \vee R\sigma}$$
    - *where A1 is eligible and* σ *is the mgu of A1 and A2*
- **Equality Factoring**
  - $$\frac{t_1 \approx u \vee t_2 \approx v \vee R}{u\sigma \neq v\sigma \vee t_1\sigma \approx v\sigma \vee R\sigma}$$
    - *where* $t_1 \approx u$ *is eligible and* σ *is the mgu of* t_1 *and* t_2

Contd...

- **Reflexivity Resolution**

$$\frac{t_1 \neq t_2 \vee R}{R\sigma}$$

  - *where* $t_1 \neq t_2$ *is eligible and* σ *is the mgu of* t_1 *and* t_2

- **Resolution**

$$\frac{A_1 \vee R_1 \qquad \neg A_2 \vee R_2}{R_1\sigma \vee R_2\sigma}$$

  - *where* A_1 *and* $\neg A_2$ *are eligible and* σ *is the mgu of* A_1 *and* A_2

Contd...

- **Ordered Paramodulation**

$$\frac{t_1 \approx u \vee R_1 \qquad L[\,t_2\,] \vee R_2}{L[\,u\,]\sigma \vee R_1\sigma \vee R_2\sigma}$$

  - *where t_2 is not a variable and where $t_1 \approx u$ and the literal $L[\,t_2\,]$ are eligible, σ is the mgu of t_1 and t_2 and $u \geq t_1$*

The Decision Procedure

- Clausal Normal Form Translation
- Guarded Clauses
- Preservation of Guardedness
- Complexity

Clausal Normal Form Translation

- Let $F=\{F_1,...,F_n\}$ (negation normal form)
  - If **F** is a formula in *F* containing $\forall\bar{x}(\neg G \vee H)$ then
    - Add $\forall\overline{x\,y}(\neg G \vee \neg\alpha(\bar{y}) \vee H)$ to *F*
    - Replace the indicated sub−formula in F by $\alpha(\bar{y})$

Assuming that:

- $\bar{y}$ is the set of variables occur in G
- But not in $\bar{x}$
- α is a new predicate name that does not occur in *F*

Examples.

- **A Guarded formula**

  $\exists x\ (n(x) \wedge \forall y[\neg a(x,y) \vee$

  $\forall z\{\neg p(x,z) \vee \exists x(a(x,z) \wedge (\neg b(z,z) \vee \neg c(x,x)))\}])$

  The Structural transformation:

  $\exists x\ [n(x) \wedge \alpha(x)],$

  $\forall x,y[\neg a(x,y) \vee \alpha(x) \vee \beta(x)],$

  $\forall x,z[\neg p(x,z) \vee \neg \beta(x) \vee$

  $\exists x(a(x,z) \wedge (\neg b(z,z) \vee \neg c(x,x)))]$

Contd...

- **The Structural transformation:**

  $\exists x\ [n(x) \wedge \alpha(x)],$

  $\forall x,y[\neg a(x,y) \vee \alpha(x) \vee \beta(x)],$

  $\forall x,z[\neg p(x,z) \vee \neg \beta(x) \vee$

  $\exists x(a(x,z) \wedge (\neg b(z,z) \vee \neg c(x,x)))]$

  Skolemization yields

  $n(c) \wedge \alpha(c),$

  $\forall x,y[\neg a(x,y) \vee \neg \alpha(x) \vee \beta(x)],$

  $\forall x,z[\neg p(x,z) \vee \neg \beta(x) \vee (a(fxz,z) \wedge (\neg b(z,z) \vee \neg c(fxz,fxz)))]$

Contd...

- **Skolemization yields**

  $n(c) \wedge \alpha(c)$,

  $\forall x,y[\neg a(x,y) \vee \neg \alpha(x) \vee \beta(x)]$,

  $\forall x,z[\neg p(x,z) \vee \neg \beta(x) \vee (a(fxz,z) \wedge (\neg b(z,z) \vee \neg c(fxz,fxz)))]$

  Finally Clausification,

  $n(c)$

  $\alpha(c)$

  $\neg a(x,y) \vee \neg \alpha(x) \vee \beta(x)$

  $\neg p(x,z) \vee \neg \beta(x) \vee a(f(x,z),z)$

  $\neg p(x,z) \vee \neg \beta(x) \vee \neg b(z,z) \vee \neg c(fxz,fxz)$

Guarded Clauses

- A simple clause C is called guarded if:
  - C is a positive, non−functional, single−variable clause; or
  - every functional subterm in C contains all variables of C, and, if C is non−ground, C contains a non functional negative literal(guard), containing all the variables of C.

A set of clauses is called guarded if all its clauses guarded.

Examples GC

- $p(0,s(0)) \vee c \neq d \vee q(s(0),f(0,0))$
- $p(x,x) \vee q(x)$
- $\neg p(y,x) \vee \neg \underline{q(x,y,y)} \vee r(x+y,x-y,x)$
- $\neg \underline{p(y,x)} \vee \neg q(x,y,y)$
- $\underline{x \neq y} \vee x \approx (x+y)$

where ***guards*** have been underlined.

Contd...

- Non Guarded Clauses
  - $\neg e(x) \vee e(s(s(x)))$ (not simple)
  - $\neg p(x) \vee \neg q(y) \vee r(x,y)$ (no guard)
  - $\neg p(f(x,y)) \vee p(x,y)$ (no guard)
  - $\neg p(x,y) \vee p(f(x),y)$ (not covering)
  - $\neg p(x,y) \vee p(0,g(x,y))$ (constant,but non−ground)

Theorem

- The number of Different GC over a finite signature has double exponential upper bound in the size of signature.
  - GC over given signature is bounded by
    - $n^s = n^{(a*a)+a+1}$
  - then the no. simple literals is at most:
    - $l=2n^{(a*a)+a+1}$
  - No. of GC from non−repeated literals is bounded by:
    - $c=2^l$

Preservation of Guardedness

- **Lemma 1.**

  Let L_1,L_2 be two literals of GC. Assume that L_2 contains a non ground functional term, whileL_1 does not. Then $L_2 > L_1$

- **Lemma 2.**

  With > and Σ described earlier, a literal in a clause is eligible for an inference only if it contains all the variables of the clause.

Contd...

- **Lemma 3**

  Let σ be the mgu of two non equational atoms $p(t_1,...,t_n)$ and $p(u_1,...,u_n)$. Then $p(t_1,...,t_n)\sigma$ is also simple.

- **Lemma 4**

  Let A and B be simple atoms such that:

  - every variable in B is also in A
  - every variable in a functional term of B is also in functional term of A
  - Every functional term of B contains all the variables of A

Contd...

Then for any substitution σ

- if Aσ is simple, then Bσ is simple
- every variable of Bσ occurs in Aσ
- every variable occurring in a functional term of Bσ occurs in functional term of Aσ
- every functional term of Bσ contains all the variables of Aσ

Contd...

- **Lemma 5**
  - A factor of guarded clause is guarded.
- **Lemma 6**

  An equality factor of guarded clause is guarded.
- **Lemma 7**

  A clause obtained by reflexivity resolution from guarded clause, is guarded.

Contd...

- **Lemma 8**

  A resolvent of two guarded clauses is guarded.

- **Lemma 9**

  Any clause obtained by a superposition inference from two guarded clauses is guarded.

Contd...

- **Theorem 1**

  Let Σ and $>$ be as specified. For all the inferences of the ordered paramodulation calculus, if the premises are guarded, so is the conclusion.

- **Theorem 2**

  The fragment of guarded clauses is decidable by ordered paramodulation.

Complexity

- The superposition decision procedure can be implemented in 2EXPTIME (in the size of signature).

Summary

- Ordered Paramodulation with selection is a decision procedure for the GF with equality.
- The procedure decides the class of guarded clauses which is a proper superclass of GF with equality.
- The worst case time complexity of the decision procedure is double−exponential, which is optimal given with 2−EXPTIME−complete.

Contd...

- GC with deep terms, decidable without equality can become undecidable in equational case.

Conclusion

- Hope for obtaining usable decision procedures from the discussed results
- Theoretical optimality result and experimental evidence obtained from theorem proving techniques supports the hope
- Easy to configure SP into an optimal decision procedure

Contd...

- Saturation−based theorem proving is sufficient to solve the problems without dealing with technically difficult proof−theoretic arguments.