Using FOL Theorem Provers

Marijn J.H. Heule

Carnegie Mellon University

Introduction

Herbrand's Theorem

Using FOL Theorem Provers

Introduction

Herbrand's Theorem

Using FOL Theorem Provers

Introduction: Recap FOL

Undecidable

- No single algorithm decides on all formulas
- E.g., no decision procedure for non-linear integer arithmetic

Introduction: Recap FOL

Undecidable

- No single algorithm decides on all formulas
- E.g., no decision procedure for non-linear integer arithmetic

Refutationally complete

- Decision procedures for valid/unsatisfiable formulas
- E.g., pure FOL formulas with universal sentences
- The algorithm could run forever on satisfiable formulas

Introduction: Recap FOL

Undecidable

- No single algorithm decides on all formulas
- E.g., no decision procedure for non-linear integer arithmetic

Refutationally complete

- Decision procedures for valid/unsatisfiable formulas
- E.g., pure FOL formulas with universal sentences
- The algorithm could run forever on satisfiable formulas

Decidable

- Decision procedures for all formulas in a FOL fragment
- E.g., Presburger arithmetic, uninterpreted functions

Introduction: Differences FOL and Propositional Logic

Decidable

- Propositional logic is decidable
- Only some fragments of FOL are decidable

Introduction: Differences FOL and Propositional Logic

Decidable

- Propositional logic is decidable
- Only some fragments of FOL are decidable

Compexity of resolution

- Variable elimination determines satisfiability in Prop
- Clauses can produce infinitely many resolvents in FOL
- E.g. $\forall x.\, \neg P(x) \vee P(f(x))$

Introduction: Differences FOL and Propositional Logic

Decidable

- Propositional logic is decidable
- Only some fragments of FOL are decidable

Compexity of resolution

- Variable elimination determines satisfiability in Prop
- Clauses can produce infinitely many resolvents in FOL
- E.g. $\forall x.\, \neg P(x) \vee P(f(x))$

Properties of resolution

- In Prop, resolving on multiple clashing pairs $\longrightarrow$ tautology
- In FOL, literals may be merged
- E.g. $\forall y.\, \neg S(a,y) \vee \neg S(y,y)$ and $\forall z.\, S(z,z) \vee S(a,z)$

Introduction

Herbrand's Theorem

Using FOL Theorem Provers

The Big Picture

We turn a FOL formula of universal sentences into a (typically infinite) propositional formula and have the following:

- The propositional formula is satisfiable iff every finite subset of the formula is satisfiable (Compactness Lemma).
- The satisfying assignment of the (possibly infinite) propositional formula can be turned into a (possibly infinite) model of the FOL formula (Herbrand's Theorem).
- If a finite set of the propositional formula is unsatisfiable, then the FOL formula is unsatisfiable (Lifting Lemma).

Compactness Lemma (Propositional Logic)

Theorem (Compactness)

Let τ be any partial truth assignment, and suppose that every finite subset of Γ is satisfied by a truth assignment that extends τ. Then for every propositional variable p_i, either every finite subset of Γ is satisfied by a truth assignment that extends $\tau[p_i \mapsto \top]$ or every finite subset of Γ is satisfied by a truth assignment that extends $\tau[p_i \mapsto \bot]$.

Compactness Lemma (Propositional Logic)

Theorem (Compactness)

Let τ be any partial truth assignment, and suppose that every finite subset of Γ is satisfied by a truth assignment that extends τ. Then for every propositional variable p_i, either every finite subset of Γ is satisfied by a truth assignment that extends $\tau[p_i \mapsto \top]$ or every finite subset of Γ is satisfied by a truth assignment that extends $\tau[p_i \mapsto \bot]$.

Proof.

Suppose otherwise. Then there is a finite subset Γ_0 of Γ that is not satisfied by any truth assignment that extends $\tau[p_i \mapsto \top]$ and there is a finite subset Γ_1 of Γ that is not satisfied by any truth assignment that extends $\tau[p_i \mapsto \bot]$. Then there is no truth assignment extending τ that satisfies $\Gamma_0 \cup \Gamma_1$, because any such truth assignment has to assign $\top$ or $\bot$ to p_i. $\square$

Herbrand's Theorem

Theorem

Let Γ be a set of universal first-order formulas in a language that contains at least one constant. Suppose every finite set of ground instances of formulas in Γ is satisfiable as a set of propositional formulas. Then there exists a model of Γ.

Herbrand's Theorem

Theorem

Let Γ be a set of universal first-order formulas in a language that contains at least one constant. Suppose every finite set of ground instances of formulas in Γ is satisfiable as a set of propositional formulas. Then there exists a model of Γ.

Theorem (Herbrand's Theorem)

Let $A = \forall x_1 \ldots x_n.\ A_1 \wedge \ldots \wedge A_m$ be a clause normal form formula with constant and function symbols from Σ. Let Σ' be the set of ground terms that can be made from symbols in Σ.

A is unsatisfiable if and only if there is a finite set Γ where:

- *Each element in Γ is a clause $A_i[t_1/x_1, \ldots t_n/x_n]$ where $1 \leq i \leq m$ and $t_1 \ldots t_n \in \Sigma'$.*
- *If each distinct literal in Γ is interpreted as a unique prop variable, then Γ is unsatisfiable in prop logic.*

Herbrand Model

Herbrand universe: set of all ground terms

Example

Consider $\Gamma := \forall x, y.\ (f(x) \neq c) \wedge (f(x) \neq f(y) \vee x = y)$.

The Herbrand universe of Γ is $|\mathfrak{M}| := \{c, f(c), f(f(c), \dots\}$.

The Herbrand universe can be infinite while Γ is finite.

Herbrand Model

Herbrand universe: set of all ground terms

Example

Consider $\Gamma := \forall x, y.\ (f(x) \neq c) \wedge (f(x) \neq f(y) \vee x = y)$.

The Herbrand universe of Γ is $|\mathfrak{M}| := \{c, f(c), f(f(c), \dots\}$.

The Herbrand universe can be infinite while Γ is finite.

Herbrand model: interpret each constant/function as itself

- $c^{\mathfrak{M}}$ is interpreted as c
- $f^{\mathfrak{M}}(t_1, \dots, t_n)$ is interpreted as $f(t_1, \dots, t_n)$

Transforming FOL to Propositional Logic

Example

Let $\Gamma := \forall x, y.\ \neg R(f(x), c) \wedge \big(\neg R(f(x), f(y)) \vee R(x, y)\big)$.

The Herbrand universe of Γ is $|\mathfrak{M}| := \{c, f(c), f(f(c)), \dots\}$.
Γ can be transformed into the (infinite) propositional formula:

$$\neg p_{R(f(c),c)} \wedge \big(\neg p_{R(f(c),f(c))} \vee p_{R(c,c)}\big) \wedge \qquad x = c, y = c$$
$$\neg p_{R(f(f(c)),c)} \wedge \big(\neg p_{R(f(f(c)),f(y))} \vee p_{R(f(c),c)}\big) \wedge \qquad x = f(c), y = c$$
$$\neg p_{R(f(c),c)} \wedge \big(\neg p_{R(f(c),f(f(c)))} \vee p_{R(c,f(c))}\big) \wedge \qquad x = c, y = f(c)$$
$$\dots$$

Transforming FOL to Propositional Logic

Example

Let $\Gamma := \forall x, y.\ \neg R(f(x), c) \wedge \big(\neg R(f(x), f(y)) \vee R(x, y)\big)$.

The Herbrand universe of Γ is $|\mathfrak{M}| := \{c, f(c), f(f(c)), \dots\}$.
Γ can be transformed into the (infinite) propositional formula:

$$\neg p_{R(f(c),c)} \wedge \big(\neg p_{R(f(c),f(c))} \vee p_{R(c,c)}\big) \wedge \qquad x = c, y = c$$
$$\neg p_{R(f(f(c)),c)} \wedge \big(\neg p_{R(f(f(c)),f(y))} \vee p_{R(f(c),c)}\big) \wedge \qquad x = f(c), y = c$$
$$\neg p_{R(f(c),c)} \wedge \big(\neg p_{R(f(c),f(f(c)))} \vee p_{R(c,f(c))}\big) \wedge \qquad x = c, y = f(c)$$
$$\dots$$

The equality relation needs special attention

- Replace it with a new relation, say $E(x, y)$
- Enforce symmetry, transitivity, and congruence

Proof of Herbrand's Theorem

Suppose every finite set of ground instances of the formulas in Γ is satisfiable as a set of propositional formulas. Then, by the compactness theorem for propositional logic, there is a truth assignment τ that satisfies all the ground instances of the formulas in Γ.

Proof of Herbrand's Theorem

Suppose every finite set of ground instances of the formulas in Γ is satisfiable as a set of propositional formulas. Then, by the compactness theorem for propositional logic, there is a truth assignment τ that satisfies all the ground instances of the formulas in Γ.

We look to τ to determine the truth of the atomic formulas in the language of Γ. Let $\mathfrak{M}$ be the Herbrand model of Γ.
For every R, we define $R^{\mathfrak{M}}$ to hold iff $\tau(R(t_1, \ldots, t_n))$ is true.
In other words, first-order evaluation in the model $\mathfrak{M}$ is the same as propositional evaluation under the truth assignment τ.

Proof of Herbrand's Theorem

Suppose every finite set of ground instances of the formulas in Γ is satisfiable as a set of propositional formulas. Then, by the compactness theorem for propositional logic, there is a truth assignment τ that satisfies all the ground instances of the formulas in Γ.

We look to τ to determine the truth of the atomic formulas in the language of Γ. Let $\mathfrak{M}$ be the Herbrand model of Γ.
For every R, we define $R^{\mathfrak{M}}$ to hold iff $\tau(R(t_1, \ldots, t_n))$ is true.
In other words, first-order evaluation in the model $\mathfrak{M}$ is the same as propositional evaluation under the truth assignment τ.

Since the universe of $\mathfrak{M}$ consists of ground terms, a formula $\forall x_1, \ldots, x_n.\ A$ in Γ is true in $\mathfrak{M}$ iff for every $t_1, \ldots, t_n$, A is true under the assignment $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$. By the definition of $\mathfrak{M}$, this is the case iff $\tau(A(t_1, \ldots, t_n))$ is true.
This holds as τ satisfies every closed instance of a formula in Γ.

Lifting Lemma

If a finite set of the propositional formula is unsatisfiable, then the FOL formula is unsatisfiable.

Lifting Lemma

If a finite set of the propositional formula is unsatisfiable, then the FOL formula is unsatisfiable.

Lemma (Lifting Lemma)

Let $A = \forall x_1, \ldots, x_n.\ A_1 \wedge \ldots \wedge A_m$ *be a clause normal form formula with constant and function symbols from* Σ*. Let* Σ' *be the set of closed terms that can be made from symbols in* Σ*.*

Let Γ *be a set where each element is a clause* $A_i[t_1/x_1, \ldots t_n/x_n]$ *where* $1 \leq i \leq m$ *and* $t_1 \ldots t_n \in \Sigma'$*.*

If each distinct literal in Γ *is interpreted as a unique propositional variable, then any propositional resolution refutation of* Γ *can be transformed into a first-order resolution refutation of* A*.*

Introduction

Herbrand's Theorem

Using FOL Theorem Provers

Using FOL Theorem Provers: Aunt Agatha

Someone who lives in Dreadbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadbury Mansion, and are the only people who live therein. A killer always hates his victim, and is never richer than his victim. Charles hates no one that Aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone Aunt Agatha hates. No one hates everyone. Agatha is not the butler.

Who killed Aunt Agatha?

Using FOL Theorem Provers: Aunt Agatha

Someone who lives in Dreadbury Mansion killed Aunt Agatha. Agatha, the butler, and Charles live in Dreadbury Mansion, and are the only people who live therein. A killer always hates his victim, and is never richer than his victim. Charles hates no one that Aunt Agatha hates. Agatha hates everyone except the butler. The butler hates everyone not richer than Aunt Agatha. The butler hates everyone Aunt Agatha hates. No one hates everyone. Agatha is not the butler.

Who killed Aunt Agatha?

```
def aunt_agatha_hypotheses : List FOForm := [
  fo!{∃ x. lives_at_dreadbury(%x) ∧ killed(%x, agatha)},
  fo!{∀ x. lives_at_dreadbury(%x) ↔ (%x = agatha ∨ %x = butler ∨ %x = charles)},
  fo!{∀ x. ∀ y. killed(%x, %y) → hates(%x, %y)},
  fo!{∀ x. ∀ y. killed(%x, %y) → ¬ richer(%x, %y)},
  fo!{∀ x. hates(charles, %x) → ¬ hates(agatha, %x)},
  fo!{∀ x. ¬ hates(agatha, %x) ↔ %x = butler},
  fo!{∀ x. ¬ richer(%x, agatha) → hates(butler, %x)},
  fo!{∀ x. hates(agatha, %x) → hates(butler, %x)},
  fo!{∀ x. ∃ y. ¬ hates(%x, %y)},
  fo!{¬ agatha = butler}
]
```

Using FOL Theorem Provers: Aunt Agatha in Lean

Examples/using_first_order_theorem_provers/aunt_agatha.lean

```
def aunt_agatha_hypotheses : List FOForm := [
  fo!{∃ x. lives_at_dreadbury(%x) ∧ killed(%x, agatha)},
  fo!{∀ x. lives_at_dreadbury(%x) ↔ (%x = agatha ∨ %x = butler ∨ %x = charles)},
  fo!{∀ x. ∀ y. killed(%x, %y) → hates(%x, %y)},
  fo!{∀ x. ∀ y. killed(%x, %y) → ¬ richer(%x, %y)},
  fo!{∀ x. hates(charles, %x) → ¬ hates(agatha, %x)},
  fo!{∀ x. ¬ hates(agatha, %x) ↔ %x = butler},
  fo!{∀ x. ¬ richer(%x, agatha) → hates(butler, %x)},
  fo!{∀ x. hates(agatha, %x) → hates(butler, %x)},
  fo!{∀ x. ∃ y. ¬ hates(%x, %y)},
  fo!{¬ agatha = butler}
]
```

Did the butler kill aunt Agatha?

Using FOL Theorem Provers: Raymond Smullyan

In an article called "The Asylum of Doctor Tarr and Professor Fether," Raymond Smullyan tells of an investigation of 11 insane asylums by Inspector Craig of Scotland Yard.

In each of these asylums, every inhabitant x is either a doctor (Doctor(x)) or a patient ($\neg$Doctor(x)), and every inhabitant is either sane (Sane(x)) or insane ($\neg$Sane(x)).

The sane inhabitants are totally sane and the insane inhabitants are totally insane, in the following sense:
for any proposition P,
a sane inhabitant believes P if and only if P is true,
and an insane inhabitant believes P if and only if P is false.

Using FOL Theorem Provers: The 8th Asylum

1. Some of the inhabitants are teachers of other inhabitants. Each inhabitant has at least one teacher.

Using FOL Theorem Provers: The 8th Asylum

1. Some of the inhabitants are teachers of other inhabitants. Each inhabitant has at least one teacher.
   $\forall x.\ \exists y.\ \text{Teaches}(y, x)$
2. There is one inhabitant who trusts all the patients but does not trust any of the doctors.

Using FOL Theorem Provers: The 8th Asylum

1. Some of the inhabitants are teachers of other inhabitants. Each inhabitant has at least one teacher.
   $\forall x.\ \exists y.\ \mathsf{Teaches}(y, x)$
2. There is one inhabitant who trusts all the patients but does not trust any of the doctors.
   $\exists x.\ \forall y.\ \neg\mathsf{Doctor}(y) \leftrightarrow \mathsf{Trusts}(x, y)$
3. No inhabitant x is willing to be a teacher of an inhabitant y unless x believes that y trusts himself.

Using FOL Theorem Provers: The 8th Asylum

1. Some of the inhabitants are teachers of other inhabitants. Each inhabitant has at least one teacher.
   $\forall x.\ \exists y.\ \mathsf{Teaches}(y, x)$
2. There is one inhabitant who trusts all the patients but does not trust any of the doctors.
   $\exists x.\ \forall y.\ \neg\mathsf{Doctor}(y) \leftrightarrow \mathsf{Trusts}(x, y)$
3. No inhabitant x is willing to be a teacher of an inhabitant y unless x believes that y trusts himself.
   $\forall x, y.\ \mathsf{Teaches}(x, y) \rightarrow (\mathsf{Sane}(x) \leftrightarrow \mathsf{Trusts}(y, y))$
4. For any inhabitant x there is an inhabitant y who trusts all and only those inhabitants who have at least one teacher who is trusted by x.

Using FOL Theorem Provers: The 8th Asylum

1. Some of the inhabitants are teachers of other inhabitants. Each inhabitant has at least one teacher.
   $\forall x.\ \exists y.\ \mathsf{Teaches}(y, x)$
2. There is one inhabitant who trusts all the patients but does not trust any of the doctors.
   $\exists x.\ \forall y.\ \neg\mathsf{Doctor}(y) \leftrightarrow \mathsf{Trusts}(x, y)$
3. No inhabitant x is willing to be a teacher of an inhabitant y unless x believes that y trusts himself.
   $\forall x, y.\ \mathsf{Teaches}(x, y) \rightarrow (\mathsf{Sane}(x) \leftrightarrow \mathsf{Trusts}(y, y))$
4. For any inhabitant x there is an inhabitant y who trusts all and only those inhabitants who have at least one teacher who is trusted by x.
   $\forall x.\ \exists y.\ \forall z.\ \mathsf{Trusts}(y, z) \leftrightarrow \exists w.\ \mathsf{Teaches}(w, z) \wedge \mathsf{Trusts}(x, w)$

Using FOL Theorem Provers: The 8th Asylum in Lean

Examples/using_first_order_theorem_provers/asylum_eight.lean

```
def asylum_eight_hypotheses : List FOForm := [
  fo!{∀ x. ∃ y. Teaches(%y, %x)},
  fo!{∀ x. ∀ y. Teaches(%x, %y) → (Sane(%x) ↔ Trusts(%y, %y))},
  fo!{∀ x. ∃ y. ∀ z. Trusts(%y, %z) ↔ ∃ w. Teaches(%w, %z) ∧ Trusts(%x, %w)},
  fo!{∃ x. ∀ y. ¬ Doctor(%y) ↔ Trusts(%x, %y)}
]
```

```
def asylum_eight_conclusion :=
  fo!{∃ x. Doctor(%x) ↔ ¬ Sane(%x)}

#eval (do
  discard <| callVampireTptp asylum_eight_hypotheses asylum_eight_conclusion
    (verbose := true)
  : IO Unit)
```

Try at Home

Refute the 8th Asylum using resolution

- Skolemize the formula
- Turn it into CNF
- Negate the conclusion

Encode the Last Asylum in Lean

- The description is the Section 17.3 of the textbook
- Solve it using Vampire