15-780: Graduate AI

Lecture 3. FOL proofs; SAT

Geoff Gordon (this lecture)
Tuomas Sandholm
TAs Byron Boots, Sam Ganzfried
Admin
HW1

- *Out today*
- *Due Tue, Feb. 3 (two weeks)*
  - *hand in hardcopy at beginning of class*
- *Covers propositional and FOL*
- *Don’t leave it to the last minute!*
Collaboration policy

- OK to discuss general strategies
- What you hand in must be your own work
  - written with no access to notes from joint meetings, websites, etc.
- You must acknowledge all significant discussions, relevant websites, etc., on your HW
Late policy

- You have 3 late days in total to split across all HWs
  - these account for conference travel, holidays, illness, or any other reasons
- After late days, 75% for next day, 50% for next, 0% thereafter (but still must turn in)
- Day = 24 hrs, HWs due at 10:30AM
Office hours

- Office hours start this week (see website for times)

- But, I have a conflict this week due to admissions; let me know by email if there is demand, and if so I can reschedule
Matlab tutorial

- Thu 1/22, 4–5PM, Wean Hall 5409
Review
In propositional logic

- Compositional semantics, structural induction
- Proof trees, proof by contradiction
- Inference rules (e.g., resolution)
- Soundness, completeness
- Horn clauses
- Nonmonotonic logic
In FOL

- Compositional semantics
  - objects, functions, predicates
  - terms, atoms, literals, sentences
  - quantifiers, free/bound variables
  - models, interpretations
- Generalized de Morgan’s law
- Skolemization, CNF
Project Ideas
Traffic insanity
Sensor planning

- Plan a path for this robot so that it gets a good view of an object as fast as possible
Mini-robots

- Do something cool w/ Lego Mindstorms
  - plan footstep placements
  - plan how to grip objects
Poker
Poker

- Minimax strategy for heads-up poker = solving linear program
- 1-card hands, 13-card deck: 52 vars, instantaneous
- RI Hold’Em: ~1,000,000 vars
  - 2 weeks / 30GB (exact sol, CPLEX)
  - 40 min / 1.5GB (approx sol)
- TX Hold’Em: ??? (up to $10^{17}$ vars or so)
Poker

- Learning by repeated play
  - we’ll discuss learning algorithms later
- Possibly state-of-the-art for 2 players
- We don’t know another feasible approach for 3 or more players
- Project: pick a poker domain, compare several learning algorithms and/or other solution methods
Understand the web

- Write a probabilistic knowledge base describing a portion of the web
- Learn parameters of the model
Proofs in FOL
FOL is special

- Despite being much more powerful than propositional logic, there is still a sound and complete inference procedure for FOL.
- Almost any significant extension breaks this property.
- This is why FOL is popular: very powerful language with a sound & complete inference procedure.
Proofs

- **Proofs by contradiction work as before:**
  - *add* $\neg S$ to $KB$
  - *put in CNF*
  - *run resolution*
  - *if we get an empty clause, we’ve proven $S$ by contradiction*

- **But, CNF and resolution have changed**
Generalizing resolution

- **Propositional:** \((-a \lor b) \land a \models b\)
- **FOL:**
  
  \((-\text{man}(x) \lor \text{mortal}(x)) \land \text{man}(\text{Socrates})\)
  
  \((-\text{man}(\text{Socrates}) \lor \text{mortal}(\text{Socrates})) \land \text{man}(\text{Socrates})\)
  
  \(\models \text{mortal}(\text{Socrates})\)

- **Difference:** had to substitute \(x \rightarrow \text{Socrates}\)
Universal instantiation

- What we just did is UI:

\[(\neg \text{man}(x) \lor \text{mortal}(x))\]
\[\models (\neg \text{man}(\text{Socrates}) \lor \text{mortal}(\text{Socrates}))\]

- Works for \(x \rightarrow\) any ground term

\[(\neg \text{man}(\text{uncle}(\text{student}(\text{Socrates})))) \lor \text{mortal}(\text{uncle}(\text{student}(\text{Socrates}))))\]

- For proofs, need a good way to find useful instantiations
Substitution lists

- List of variable $\rightarrow$ value pairs
- Values may contain variables (leaving flexibility about final instantiation)
- But, no LHS may be contained in any RHS
  - i.e., applying substitution twice is the same as doing it once
- E.g., $x \rightarrow Socrates, y \rightarrow LCA(Socrates, z)$

$LCA = \text{last common advisor}$
Unification

- Two FOL terms **unify** with each other if there is a substitution list that makes them syntactically identical
- \( \text{man}(x), \text{man}(\text{Socrates}) \) unify using the substitution \( x \rightarrow \text{Socrates} \)
- **Importance**: purely syntactic criterion for identifying useful substitutions
Unification examples

- $\text{loves}(x, x), \text{loves}(\text{John}, y)$ unify using $x \rightarrow \text{John}, y \rightarrow \text{John}$

- $\text{loves}(x, x), \text{loves}(\text{John}, \text{Mary})$ can’t unify

- $\text{loves}(\text{uncle}(x), y), \text{loves}(z, \text{aunt}(z))$: 
Unification examples

- \(\text{loves}(x, x), \text{loves}(\text{John}, y)\) unify using
  \(x \rightarrow \text{John}, y \rightarrow \text{John}\)

- \(\text{loves}(x, x), \text{loves}(\text{John}, \text{Mary})\) can’t unify

- \(\text{loves}(\text{uncle}(x), y), \text{loves}(z, \text{aunt}(z)):\)
  - \(z \rightarrow \text{uncle}(x), y \rightarrow \text{aunt}(\text{uncle}(x))\)
  - \(\text{loves}(\text{uncle}(x), \text{aunt}(\text{uncle}(x)))\)
Quiz

- Can we unify
  \[ \text{knows}(\text{John}, x) \quad \text{knows}(x, \text{Mary}) \]

- What about
  \[ \text{knows}(\text{John}, x) \quad \text{knows}(y, \text{Mary}) \]
Quiz

- **Can we unify**

  \[ \text{knows}(\text{John}, x) \quad \text{knows}(x, \text{Mary}) \]

  **No!**

- **What about**

  \[ \text{knows}(\text{John}, x) \quad \text{knows}(y, \text{Mary}) \]

  \[ x \rightarrow \text{Mary}, \ y \rightarrow \text{John} \]
Standardize apart

- But knows(x, Mary) is logically equivalent to knows(y, Mary)!
- Moral: standardize apart before unifying
Most general unifier

- **May be many substitutions that unify two formulas**

- **MGU is unique (up to renaming)**

- **Simple, moderately fast algorithm for finding MGU (see RN); more complex, linear-time algorithm**

First-order resolution

- Given clauses \((a \lor b \lor c), (\neg c' \lor d \lor e)\), and a substitution list \(V\) unifying \(c\) and \(c'\)

- Conclude \((a \lor b \lor d \lor e) : V\)
Example

\[
\begin{align*}
\text{rains n outside}(x) & \Rightarrow \text{wet}(x) \\
\text{wet}(x) & \Rightarrow \text{rusty}(x) \lor \text{rustproof}(x) \\
\text{robot}(x) & \Rightarrow \neg \text{rustproof}(x) \\
\text{rains} & \\
\text{guideboot}(\text{Robby}) & \\
\text{guideboot}(x) & \equiv \text{robot}(x) \lor \text{outside}(x)
\end{align*}
\]
rains n outside(x) =⇒ wet(x)
\[\neg\text{rains } n \neg\text{outside}(x) v \text{wet}(x)\]
\[\text{wet}(x) =⇒ \text{rusty}(x) v \text{rustproof}(x)\]
\[\text{rusty}(y) v \text{rusty}(y) v \text{rustproof}(y)\]
\[\text{robot}(x) =⇒ \neg\text{rustproof}(x)\]
\[\neg\text{robot}(z) v \neg\text{rustproof}(z)\]
\[\text{rains}\]
\[\text{guidebot}(\text{Robby})\]
\[\text{guidebot}(x) =⇒ \text{robot}(x) v \text{outside}(x)\]
\[\neg\text{guidebot}(a) v \text{robot}(a)\]
\[\neg\text{guidebot}(b) v \text{outside}(b)\]
\[\neg(\exists x. \text{rusty}(x))\]
\[\neg\text{rusty}(c)\]
\[1, 2 = \neg\text{robot}(\text{Robby})\]
\[1, 3 = \neg\text{outside}(\text{Robby})\]
\[4, 5 = \neg\text{rustproof}(\text{Robby})\]
\[6, 7 = \neg\text{rains } v \text{wet}(R)\]
\[8, 9 = \neg\text{wet}(\text{Robby})\]
\[10, 11 = \neg\text{wet}(\text{Robby}) v \text{rusty}(R)\]
\[12, 13 = \neg\text{rusty}(R)\]
\[14, 15 = F\]
First-order factoring

- When removing redundant literals, we have the option of unifying them first
- Given clause \((a \lor b \lor c)\), substitution \(V\)
- If \(a : V\) and \(b : V\) are the same
- Then we can conclude \((a \lor c) : V\)
Completeness

- *First-order resolution (together with first-order factoring) is sound and complete for* \( \text{FOL} \)

- *Famous theorem*
Completeness
Proof strategy

- We’ll show FOL completeness by reducing to propositional completeness
- To prove $S$, put $KB \land \neg S$ in clause form
- Turn FOL KB into propositional KBs
  - in general, infinitely many
- Check each one in order
- If any one is unsatisfiable, we will have our proof
Propositionalization

- Given a FOL KB in clause form
- And a set of terms U (for universe)
- We can propositionalize KB under U by substituting elements of U for free variables in all combinations
Propositionalization example

- \((\neg \text{man}(x) \lor \text{mortal}(x))\)
- \(\text{man}(\text{Socrates})\)
- \(\text{favorite\_drink}(\text{Socrates}) = \text{hemlock}\)
- \(\text{drinks}(x, \text{favorite\_drink}(x))\)

- \(U = \{\text{Socrates, hemlock, Fred}\}\)
Propositionalization example

- \((\neg \text{man}(\text{Socrates}) \lor \text{mortal}(\text{Socrates}))\)
  
- \((\neg \text{man}(\text{Fred}) \lor \text{mortal}(\text{Fred}))\)

- \((\neg \text{man}(\text{hemlock}) \lor \text{mortal}(\text{hemlock}))\)

- \(\text{drinks}(\text{Socrates}, \text{favorite\_drink}(\text{Socrates}))\)
  
- \(\text{drinks}(\text{hemlock}, \text{favorite\_drink}(\text{hemlock}))\)

- \(\text{drinks}(\text{Fred}, \text{favorite\_drink}(\text{Fred}))\)

- \(\text{man}(\text{Socrates}) \land\)
  
- \(\text{favorite\_drink}(\text{Socrates}) = \text{hemlock}\)
Choosing a universe

- To check a FOL KB, propositionalize it using some universe $U$
- Which universe?
**Herbrand Universe**

- **Herbrand universe** $H$ of formula $S$:
  - start with all objects mentioned in $S$
  - or synthetic object $X$ if none mentioned
  - apply all functions mentioned in $S$ to all combinations of objects in $H$, add to $H$
  - repeat
Herbrand Universe

- E.g., loves(uncle(John), Mary) yields

\[ H = \{ \text{John, Mary, uncle(John), uncle(Mary), uncle(uncle(John)), uncle(uncle(uncle(Mary))), … } \} \]
Herbrand’s theorem

- If a FOL KB in clause form is unsatisfiable
- And $H$ is its Herbrand universe
- Then the propositionalized KB is unsatisfiable for some finite $U \subseteq H$
Significance

- *This is one half of the equivalence we want:* unsatisfiable FOL KB $\Rightarrow \exists$ finite $U$. unsatisfiable propositional KB
Example

- \((\neg \text{man}(x) \lor \text{mortal}(x)) \land \text{man}(\text{uncle}(\text{Socrates}))\)
  \(\land \neg \text{mortal}(x)\)

- \(H = \{S, u(S), u(u(S)), \ldots\}\)

- If \(U = \{u(S)\}\), \(\text{PKB} = \)
  \((\neg \text{man}(u(S)) \lor \text{mortal}(u(S))) \land \text{man}(u(S)) \land \neg \text{mortal}(u(S))\)

- Resolving twice yields \(F\)
Converse of Herbrand

- A. J. Robinson proved “lifting lemma”
- Write PKB for a propositionalization of KB (under some universe)
- Any resolution proof in PKB corresponds to a resolution proof in KB
- …and, if PKB is unsatisfiable, there is a proof of F (by prop. completeness); so, lifting it shows KB unsatisfiable
Example

\[ (\neg \text{man}(u(S)) \lor \text{mortal}(u(S))) \land \text{man}(u(S)) \land \neg \text{mortal}(u(S)) \]

- We resolved on \( \text{man}(u(S)) \) yielding \( \text{mortal}(u(S)) \)

- Lifted, resolve \( \neg \text{man}(x) \) w/ \( \text{man}(u(S)) \), binding \( x \rightarrow u(S) \)
Problems w/ Herbrand & Robinson

So, FOL KB is unsatisfiable if and only if there is a subset of its Herbrand universe making PKB unsatisfiable.

I.e., if we have a way to find proofs in propositional logic, we have a way to find them in FOL.
Proofs w/ Herbrand & Robinson

- To prove $S$, put $KB \land \neg S$ in CNF: $KB'$

- Build subsets of Herbrand universe in increasing order of size: $U_1, U_2, \ldots$

- Propositionalize $KB'$ w/ $U_i$, look for proof

- If $U_i$ unsatisfiable, use lifting to get a contradiction in $KB'$

- If $U_i$ satisfiable, move on to $U_{i+1}$
How long will this take?

- If $S$ is not entailed, we will never find a contradiction
- In this case, if $H$ infinite, we’ll never stop
- So, entailment is semidecidable
  - equivalently, entailed statements are recursively enumerable
Variation

- Restrict semantics so we only need to check one finite propositional KB

- **Unique names**: objects with different names are different (John $\neq$ Mary)

- **Domain closure**: objects without names given in KB don’t exist

- Restrictions also make entailment, validity feasible
Who? What? Where?
Wh-questions

- We’ve shown how to answer a question like “is Socrates mortal?”
- What if we have a question whose answer is not just yes/no, like “who killed JR?” or “where is my robot?”
- Simplest approach: prove $\exists x. \text{killed}(x, \text{JR})$, hope the proof is constructive
Answer literals

- Simple approach doesn’t always work
- Instead of \( \neg S(x) \), add \( (\neg S(x) \lor \text{answer}(x)) \)
- If there’s a contradiction, we can eliminate \( \neg S(x) \) by resolution and unification, leaving \( \text{answer}(x) \) with \( x \) bound to a value that causes a contradiction
Example

\( \text{kills (Jack, Cat)} \lor \text{kills (Curiosity, Cat)} \)

\( \neg \text{kills (Jack, x)} \)
\[ \text{kills (Jack, Cat)} \lor \text{kills (Curiosity, Cat)} \]
\[ \neg \text{kills (Jack, x)} \]
\[ \neg \text{kills (x, Cat)} \]

1.3 \[ x \rightarrow \text{Jack} \equiv \text{kills (Curiosity, Cat)} \]
3, 4 \[ x \rightarrow \text{Curiosity} \equiv F \]
\[ \neg \text{kills (x, Cat)} \lor \text{answers (x)} \]
1, 5 \[ x \rightarrow \text{Curiosity} \equiv \text{kills (Jack, Cat)} \lor \text{answers (Curiosity)} \]
2, 6 \[ x \rightarrow \text{Cat} \equiv \text{answers (Curiosity)} \]
FOL
Extensions
Equality

- **Paramodulation** is sound and complete for FOL+equality (see RN)
- *Or, resolution + axiom schema*
Second order logic

- SOL adds quantification over predicates
- E.g., principle of mathematical induction:
  - $\forall P. P(0) \land (\forall x. P(x) \Rightarrow P(S(x)))$
    $\Rightarrow \forall x. P(x)$
- There is no sound and complete inference procedure for SOL (Gödel’s famous incompleteness theorem)
Others

- Temporal logics ("P(x) will be true at some time in the future")
- Modal logics ("John believes P(x)")
- Nonmonotonic FOL
- First-class functions (lambda operator, application)
- ...
Using FOL
Knowledge engineering

- *Identify relevant objects, functions, and predicates*
- *Encode general background knowledge about domain (reusable)*
- *Encode specific problem instance*
- *Pose queries (is \( P(x) \) true? Find \( x \) such that \( P(x) \))*
Common themes

- RN identifies many common idioms and problems for knowledge engineering
- Hierarchies, fluents, knowledge, belief, ...
- We’ll look at a couple
Taxonomies

- $isa(Mammal, Animal)$
- $disjoint(Animal, Vegetable)$
- $partition(\{Animal, Vegetable, Mineral, Intangible\}, Everything)$
Inheritance

- Transitive: $\text{isa}(x, y) \land \text{isa}(y, z) \Rightarrow \text{isa}(x, z)$

- Attach properties anywhere in hierarchy
  - $\text{isa}(\text{Pigeon, Bird})$
  - $\text{isa}(x, \text{Bird}) \Rightarrow \text{flies}(x)$
  - $\text{isa}(x, \text{Pigeon}) \Rightarrow \text{gray}(x)$

- So, $\text{isa}(\text{Tweety, Pigeon})$ tells us Tweety is gray and flies
Physical composition

- \text{partOf}(\text{Wean4625}, \text{WeanHall})
- \text{partOf}(\text{water37}, \text{water})
- \textit{Note distinction between mass and count nouns: any partOf a mass noun is also an example of that same mass noun}
Fluents

- Fluent = property that changes over time
  - $at(Robot, \text{Wean4623}, 11\text{AM})$
- Actions change fluents
- Fluents chain together to form possible worlds
- $at(x, p, t) \land \text{adj}(p, q) \Rightarrow poss(go(x, p, q), t) \land at(x, q, result(go(x, p, q), t))$
Frame problem

- Suppose we execute an unrelated action (e.g., talk(Professor, FOL))
- Robot shouldn’t move:
  - if at(Robot, Wean4623, t), want at(Robot, Wean4623, result(talk(Professor, FOL)))
- But we can’t prove it using tools described so far!
The frame problem is that it’s a pain to list all of the things that don’t change when we execute an action

Naive solution: frame axioms
  - for each fluent, list actions that can’t change fluent
  - $\text{KB size: } O(AF)$ for $A$ actions, $F$ fluents
Frame problem

- Better solution: successor-state axioms

- For each fluent, list actions that can change it (typically fewer): if go(x, p, q) is possible,
  
  \[ at(x, q, \text{result}(a, t)) \iff a = \text{go}(x, p, q) \vee (at(x, q, t) \land a \neq \text{go}(x, q, z)) \]

- Size \(O(AE + F)\) if each action has \(E\) effects
Sadly, also necessary…

- **Debug knowledge base**
  - Severe bug: logical contradictions
  - Less severe: undesired conclusions
  - Least severe: missing conclusions

- First 2: trace back chain of reasoning until reason for failure is revealed
- Last: trace desired proof, find what’s missing