CS389L: Automated Logical Reasoning

Lecture 10: First-Order Resolution and Intro to Theories

Işıl Dillig

Review

- Last lecture: Clausal form, first-order resolution
- How to convert formulas to clausal form?
- Resolution with Implicit Factorization:

$$\frac{\begin{array}{c}\{A_1, \dots A_n, B_1, \dots, B_k\} \\ \{\neg C, D_1, \dots, D_k\}\end{array}}{\{B_1, \dots, B_k, D_1, \dots, D_k\}\sigma} \quad (\sigma = mgu(A_1, \dots A_n, C))$$

Resolution Derivation

- A clause C is derivable from a set of clauses Δ if there is a sequence of clauses $\Psi_1, \dots, \Psi_k$ terminating in C such that:
  1. $\Psi_i \in \Delta$, or
  2. Ψ_i is resolvent of some Ψ_j and Ψ_k such that $j < i \land k < i$
- Example: Consider clauses

$$\Delta = \{happy(x), sad(x)\}, \{\neg sad(y)\}$$

- Here, $\{happy(x)\}$ is derivable from Δ
- If a clause C is derivable from Δ, we write $\Delta \vdash C$

Resolution Refutation

- The derivation of the empty clause from a set of clauses Δ is called resolution refutation of Δ
- Consider set of clauses Δ:

$$\begin{array}{l}\{happy(x), sad(x)\} \\ \{\neg sad(y)\} \\ \{\neg happy(mother(joe))\}\end{array}$$

- Resolution refutation of Δ:

$$\frac{\dfrac{\{happy(x), sad(x)\} \quad \{\neg sad(y)\}}{\{happy(x)\}} \quad \{\neg happy(mother(joe))\}}{\{\ \}}$$

Refutational Soundness and Completeness

- Theorem: Resolution is sound, i.e., if $\Delta \vdash C$, then $\Delta \models C$
- Corollary: If there is a resolution refutation of Δ, Δ is indeed unsatisfiable
- Resolution with implicit factorization is also complete, i.e., if $\Delta \models C$, then $\Delta \vdash C$
- Corollary: If F is unsatisfiable, then there exists a resolution refutation of F using only resolution with factorization.
- This is called the refutational completeness of resolution.

Validity Proofs using Resolution

- How to prove validity FOL formula using resolution?
- Use duality of validity and unsatisfiability:

$$\boxed{F \text{ is valid iff } \neg F \text{ is unsatisfiable}}$$

- We will use resolution to show $\neg F$ is unsatisfiable.
- First, convert $\neg F$ to clausal form C.
- If there is a resolution refutation of C, then, by soundness, F is valid.

Example

- Everybody loves somebody. Everybody loves a lover. Prove that everybody loves everybody.
- First sentence in FOL:
- Second sentence in FOL:
- Goal in FOL:
- Thus, want to prove validity of:

$$(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.((\exists v.loves(u,v)) \to loves(w,u))) \\ \to \forall z.\forall t.loves(z,t)$$

Example, cont.

- Want to prove negation unsatisfiable:

$$\neg((\forall x.\exists y.loves(x,y) \land \forall u.\forall w.((\exists v.loves(u,v)) \to loves(w,u))) \\ \to \forall z.\forall t.loves(z,t))$$

- Convert to PNF: in NNF, quantifiers in front
- Remove inner implication:

$$\neg((\forall x.\exists y.loves(x,y) \land \forall u.\forall w.((\neg(\exists v.loves(u,v))) \lor loves(w,u))) \\ \to \forall z.\forall t.loves(z,t))$$

- Remove outer implication:

$$\neg(\neg(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.((\neg(\exists v.loves(u,v))) \lor loves(w,u) \\ \lor \forall z.\forall t.loves(z,t))$$

Example, cont.

$$\neg(\neg(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.((\neg(\exists v.loves(u,v))) \lor loves(w,u))) \\ \lor \forall z.\forall t.loves(z,t))$$

- Push innermost negation in:

$$\neg(\neg(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.(\neg loves(u,v) \lor loves(w,u)) \\ \lor \forall z.\forall t.loves(z,t))$$

- Push outermost negation in:

$$(\neg\neg(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.\neg loves(u,v)) \lor loves(w,u)) \\ \land \neg(\forall z.\forall t.loves(z,t)))$$

Example, cont.

$$(\neg\neg(\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.\neg loves(u,v) \lor loves(w,u)) \\ \land \neg(\forall z.\forall t.loves(z,t)))$$

- Eliminate double negation:

$$((\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.\neg loves(u,v) \lor loves(w,u)) \\ \land \neg(\forall z.\forall t.loves(z,t)))$$

- Push negation on second line in:

$$((\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.\neg loves(u,v) \lor loves(w,u)) \\ \land (\exists z.\exists t.\neg loves(z,t)))$$

Example, cont.

$$((\forall x.\exists y.loves(x,y) \land \forall u.\forall w.\forall v.(\neg loves(u,v) \lor loves(w,u))) \\ \land (\exists z.\exists t.\neg loves(z,t)))$$

- Now, move quantifiers to front. Restriction:

$$\exists z.\exists t.\forall x.\exists y.\forall u.\forall w.\forall v. \\ loves(x,y) \land (\neg loves(u,v) \lor loves(w,u)) \land \neg loves(z,t)$$

- Next, skolemize existentially quantified variables:

$$\forall u.\forall w.\forall v.\forall x. \\ loves(x, lover(x)) \land (\neg loves(u,v) \lor loves(w,u)) \\ \land \neg loves(joe, jane)$$

Example, cont.

$$\forall u.\forall w.\forall v.\forall x. \\ loves(x, lover(x)) \land (\neg loves(u,v) \lor loves(w,u)) \\ \land \neg loves(joe, jane)$$

- Now, drop quantifiers:

$$loves(x, lover(x)) \land (\neg loves(u,v) \lor loves(w,u)) \\ \land \neg loves(joe, jane)$$

- Convert to CNF: already in CNF!
- In clausal form:

$$\{loves(x, lover(x))\} \\ \{\neg loves(u,v), loves(w,u)\} \\ \{\neg loves(joe, jane)\}$$

Example, cont.

- Finally, we can do resolution:

$$\{loves(x, lover(x))\}$$
$$\{\neg loves(u, v), loves(w, u)\}$$
$$\{\neg loves(joe, jane)\}$$

- Resolve first and second clauses. MGU:
- Resolvent:
- Resolve new clause with third clause.
- Mgu:
- Resolvent: $\{\}$
- Thus, we have proven the formula valid.

Example II

- Use resolution to prove validity of formula:

$$\neg(\exists y.\forall z.(p(z, y) \leftrightarrow \neg\exists x.(p(z, x) \wedge p(x, z))))$$

- Convert negation to clausal form:

$$\exists y.\forall z.(p(z, y) \leftrightarrow \neg\exists x.(p(z, x) \wedge p(x, z)))$$

- To convert to NNF, get rid of $\leftrightarrow$:

$$\exists y.\forall z.(\neg p(z, y) \vee \neg\exists x.(p(z, x) \wedge p(x, z)) \wedge$$
$$(p(z, y) \vee \exists x.(p(z, x) \wedge p(x, z))))$$

Example II, cont

$$\exists y.\forall z.(\neg p(z, y) \vee \neg\exists x.(p(z, x) \wedge p(x, z)) \wedge$$
$$(p(z, y) \vee \exists x.(p(z, x) \wedge p(x, z))))$$

- Push negations in:

$$\exists y.\forall z.(\neg p(z, y) \vee \forall x.(\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$(p(z, y) \vee \exists x.(p(z, x) \wedge p(x, z))))$$

- Rename quantified variables:

$$\exists y.\forall z.(\neg p(z, y) \vee \forall x.(\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee \exists w.(p(z, w) \wedge p(w, z)))$$

Example II, cont.

$$\exists y.\forall z.(\neg p(z, y) \vee \forall x.(\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee \exists w.(p(z, w) \wedge p(w, z)))$$

- In PNF:

$$\exists y.\forall z.\exists w.\forall x.(\neg p(z, y) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, y) \vee (p(z, w) \wedge p(w, z)))$$

- Skolemize existentials:

$$\forall z.\forall x.(\neg p(z, a) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, a) \vee (p(z, f(z)) \wedge p(f(z), z)))$$

Example II, cont.

$$\forall z.\forall x.(\neg p(z, a) \vee (\neg p(z, x) \vee \neg p(x, z)) \wedge$$
$$p(z, a) \vee (p(z, f(z)) \wedge p(f(z), z)))$$

- Drop quantifiers and convert to CNF:

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

- In clausal form (with renamed variables):

$$C1: \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2: \{p(y, a), p(y, f(y))\}$$
$$C3: \{p(w, a), p(f(w), w))\}$$

Example II, cont.

$$C1: \{\neg p(z, a), \neg p(z, x), \neg p(x, z)\}$$
$$C2: \{p(y, a), p(y, f(y))\}$$
$$C3: \{p(w, a), p(f(w), w))\}$$

- Resolve $C1$ and $C2$ using factoring.
- What is the MGU for $p(z, a), p(z, x), p(x, z), p(y, a)$?
- Resolvent:

Example II, cont.

$$\begin{aligned} C1: &\ \{\neg p(z,a), \neg p(z,x), \neg p(x,z)\} \\ C2: &\ \{p(y,a), p(y, f(y))\} \\ C3: &\ \{p(w,a), p(f(w), w))\} \\ C4: &\ \{p(a, f(a))\} \end{aligned}$$

- Now, resolve $C1$ and $C3$ (using factoring).
- What is the MGU for $p(z,a), p(z,x), p(x,z), p(w,a)$?
- Resolvent:

Example II, cont.

$$\begin{aligned} C1: &\ \{\neg p(z,a), \neg p(z,x), \neg p(x,z)\} \\ C2: &\ \{p(y,a), p(y, f(y))\} \\ C3: &\ \{p(w,a), p(f(w), w))\} \\ C4: &\ \{p(a, f(a))\} \\ C5: &\ \{p(f(a), a)\} \end{aligned}$$

- Resolve $C1$ and $C5$ (using factoring).
- What is the MGU of $p(z,a), p(z,x)$ and $p(f(a),a)$?
- Resolvent:

Example II, cont.

$$\begin{aligned} C1: &\ \{\neg p(z,a), \neg p(z,x), \neg p(x,z)\} \\ C2: &\ \{p(y,a), p(y, f(y))\} \\ C3: &\ \{p(w,a), p(f(w), w))\} \\ C4: &\ \{p(a, f(a))\} \\ C5: &\ \{p(f(a), a)\} \\ C6: &\ \{\neg p(a, f(a))\} \end{aligned}$$

- Finally, resolve $C4$ and $C6$.
- Resolvent: $\{\}$
- Thus, the original formula is valid.

Resolution and First-Order Theorem Provers

- Resolution (with factorization) forms the basis of most automated first-order theorem provers.
- However, to make relational refutation more efficient, there are typically two main improvements:
  - Ordered resolution
  - Removal of useless clauses (tautology elimination, identical clause elimination etc.)
  - Built-in reasoning about equality

Motivation for First-Order Theories

- First-order logic is very powerful and very general.
- But in many settings, we have a particular application in mind and do not need the full power of first order logic.
- For instance, instead of general predicates/functions, we might only need an equality predicate or arithmetic operations.
- Also, might want to disallow some interpretations that are allowed in first-order logic.

First-Order Theories

- First-order theories: Useful for formalizing and reasoning about particular application domains
  - e.g., involving integers, real numbers, lists, arrays, . . .
- Advantage: By focusing on particular application domain, can give much more efficient, specialized decision procedures

Signature and Axioms of First-Order Theory

- A first-order theory T consists of:
  1. Signature Σ_T: set of constant, function, and predicate symbols
  2. Axioms A_T: A set of FOL sentences over Σ_T
- Σ_T formula: Formula constructed from symbols of Σ_T and variables, logical connectives, and quantifiers.
- Example: We could have a theory of heights T_H with signature $\Sigma_H : \{taller\}$ and axiom:

$$\forall x, y.\ (taller(x, y) \rightarrow \neg taller(y, x))$$

- Is $\exists x.\forall z.taller(x, z) \wedge taller(y, w)$ legal Σ_H formula?
- What about $\exists x.\forall z.taller(x, z) \wedge taller(joe, tom)$?

Axioms of First-Order Theory

- The axioms A_T provide the meaning of symbols in Σ_T.
- Example: In our theory of heights, axioms define meaning of predicate $taller$
- Specifically, axioms ensure that some interpretations legal in standard FOL are not legal in T
- Example: Consider relation constant $taller$, and $U = \{A, B, C\}$
- In FOL, possible interpretation: $I(taller) : \{\langle A, B\rangle, \langle B, A\rangle\}$
- In our theory of heights, this interpretation is not legal b/c does not satisfy axioms

Models of T

- A structure $M = \langle U, I\rangle$ is a model of theory T, or T-model, if $M \models A$ for every $A \in A_T$.
- Example: Consider structure consisting of universe $U = \{A, B\}$ and interpretation $I(taller) : \{\langle A, B\rangle, \langle B, A\rangle\}$
- Is this a model of T?
- Now, consider same U and interpretation $\langle A, B\rangle$. Is this a model?
- Suppose our theory had another axiom:

$$\forall x, y, z.\ (taller(x, y) \wedge taller(y, z) \rightarrow taller(x, z))$$

- Consider $I(taller) : \{\langle A, B\rangle, \langle B, C\rangle\}$. Is (U, I) a model?

Satisfiability and Validity Modulo T

- Formula F is satisfiable modulo T if there exists a T-model M and variable assignment σ such that $M, \sigma \models F$
- Formula F is valid modulo T if for all T-models M and variable assignments σ, $M, \sigma \models F$
- Question: How is validity modulo T different from FOL-validity?
- Answer: Disregards all structures that do not satisfy theory axioms.
- If a formula F is valid modulo theory T, we will write $T \models F$.
- Theory T consists of all sentences that are valid in T.

Questions

Consider some first order theory T:

- If a formula is valid in FOL, is it also valid modulo T?
- If a formula is valid modulo T, is it also valid in FOL?
- Counterexample: This formula is valid in "theory of heights":

$$\neg taller(x, x)$$

Equivalence Modulo T

- Two formulas F_1 and F_2 are equivalent modulo theory T if for every T-model M and for every variable assignment σ:

$$\boxed{M, \sigma \models F_1 \text{ iff } M, \sigma \models F_2}$$

- Another way of stating equivalence of F_1 and F_2 modulo T:

$$T \models F_1 \leftrightarrow F_2$$

- Example: Consider a theory $T_=$ with predicate symbol $=$ and suppose A_T gives the intended meaning of equality to $=$.
- Are $x = y$ and $y = x$ equivalent modulo $T_=$?
- Are they equivalent according to FOL semantics?
- Falsifying interpretation: $U = \{\square, \triangle\}, I(=) : \{\langle\triangle, \square\rangle\}$

Completeness of Theory

- A theory T is complete if for every sentence F, if T entails F or its negation:

$$\boxed{T \models F \text{ or } T \models \neg F}$$

- Question: In first-order logic, for every closed formula F, is either F or $\neg F$ valid?
-
- Consider $U = \{\circ, \star\}$
- Falsifying interpretation for $p(a)$:
- Falsifying interpretation for $\neg p(a)$:

Decidability of Theory

- A theory T is decidable if for every formula F, there exists an algorithm that:
  1. always terminates and answers "yes" if F is valid modulo T and
  2. terminates and answers "no" if F is not valid modulo T
- Unlike full first-order logic, many of the first-order theories we will study are decidable.
- For those that are not decidable, we are interested in fragments of that theory that are decidable.

Useful First-Order Theories

1. Theory of equality
2. Peano Arithmetic
3. Presburger Arithmetic
4. Theory of Rationals
5. Theory of Arrays