Review Session!

Finals Info

- Time: Thursday,December 12, 3:30-6:30pm
- Location: TBD

- Alternate Time: TBD
- Alternate Location: TBD

- Material covered: everything

Table of Contents

Propositional Logic

- Constants
  - Alphanumeric words
  - Start with lower case letter
  - Cannot include punctuation or other funny characters

- Compound Sentences
  - Build compound sentences using below operators

$$\neg$$

$$\wedge$$

$$\vee$$

$$\Leftrightarrow \quad \Leftarrow \quad \Rightarrow$$

Propositional Logic

- Interpretations
  - Any assignment of truth values to each of the constants.

- Valid / Contingent / Unsatisfiable
  - How do you figure these out?
    - Truth table
    - Using validities, such as deMorgan's Laws, Implication Introduction, Implication Distribution

Propositional Proofs

- Logical entailment
  - If every interpretation that satisfies the set of premises also satisfies the conclusion, then the set of premises logically entails the conclusion
- How do you show logical entailment?
  - Truth table method
  - Inference rules (MP, MT, EE, DN)
  - Axiom schemata (II, ID, CR, EQ, OQ) + MP
- Soundness & Completeness

Propositional Resolution

- Clausal form (INDO)
- Negated goal
- Resolution issues

{ p, q }	{ p, ~p }	{ ~p, q }	{ ~p, ~q }
{ p }	{ p, ~p }	{ p, q}	{ p, q}
No resolvents	{ p, ~p }	{q}	{q, ~q}
{q}		{q, q}	{p, ~p}
			{}

Relational Logic

- Term
  - Variable
  - Object constant
  - Functional term

- Sentences
  - Relational sentences
  - Logical sentences
  - Quantified sentences

Relational Logic

- Remember:
  - Terms are NOT sentences in relational logic

    $momma_1(sutin)$

  - Relational sentences can only be formed with terms.

    $loves_2(sutin, person_1(dad))$

- Relations vs. Functions
  - Relational sentences evaluate to *True* or *False*
  - Functional terms evaluate to things in the universe of discourse

Relational Logic

- Quantifiers ($\forall$ and $\exists$)

  – Order of appearance matters

  $\forall x.\exists y.loves(x,y)$

  $\exists y.\forall x.loves(x,y)$

  – A variable is bound by the "closest" available quantifier for the variable

  $\exists y.\forall x.(\exists y.q(y) \Rightarrow (\exists x.\exists y.\exists z.\forall x.p(z,x) \vee \forall u.q(x)))$

Relational Logic

- More quantifiers ($\forall$ and $\exists$)
  - You probably don't want implications in $\exists$–quantified sentences

    "sutin loves some cats"

    Universe of discourse = { all animals }

    $\exists y.(cat(y) \Rightarrow loves(sutin, y))$

    $\exists y.(cat(y) \wedge loves(sutin, y))$

Relational Logic

- More quantifiers ($\forall$ and $\exists$)
  - On the other hand, you might consider using implications for $\forall$– quantified sentences

    "sutin loves all dogs"

    Universe of discourse = { all animals }

    $\forall y.(dog(y) \Rightarrow loves(sutin, y))$

    $\forall y.(dog(y) \wedge loves(sutin, y))$

Relational Logic Semantics

- Model

  "an arbitrary set of ground, atomic sentences in which all arguments are constants"

  Sentence: $\forall y.(dog(y) \Rightarrow loves(walter, y))$

  Universe: { *fluffy*, *poopy*, *duffy*, *goofy*}

  Model 1: { *dog(goofy)*, *loves(walter, goofy)* }

  Model 2: { *dog(poopy)*, *dog(goofy)* }

Herbrand

- Herbrand universe
  – Set of all constants used in our sentences

- Herbrand base
  – Set of all ground atomic sentences formed using the constants from the Herbrand universe

- Herbrand model
  – Any subset of the Herbrand base

- Example: See problem set #2 problem #8

Herbrand

- Herbrand method
  - – Add negated conclusion to the set of premises— “satisfaction set”
  - – Loop over Herbrand interpretations and cross out the ones that do not satisfy the satisfaction set
  - – If all interpretations have been crossed out, then premises logically entail the conclusion

- When negating goal, Skolemize if the goal is universally quantified

Relational Proofs

- Universal Instantiation (UI)

$\forall y. hates(mike, y) \Rightarrow hates(mike, mother(mike))$

$\forall x. \exists y. hates(x, y) \Rightarrow \exists y. hates(woojin, y)$

$\forall x. \exists y. hates(x, y) \Rightarrow \exists y. hates(mother(y), y)$

Relational Proofs

- Existential Instantiation (EI)

$\exists y.hates(woojin, y)$ ➔ $hates(woojin, someone)$

$\exists y.hates(woojin, y)$ ➔ $hates(woojin, woojin)$

$\exists y.hates(x, y)$ ➔ $hates(x, f(x))$

$\exists y.hates(x, y)$ ➔ $hates(x, mother(x))$

$\exists y.hates(x, y)$ ➔ $hates(x, woojin)$

Relational Proofs

- Use inference rules (MP, MT, AE, AI) or axiom schemata like before…

Unification

- Pure & Impure substitutions
- Composition of substitutions
- Composable & Noncomposable substitutions
- Unifier
- Most General Unifier

Unification

- Unifier?

$$p(u,v)\ \{\ u \leftarrow y,\ x \leftarrow v\ \} = p(y,v)$$

$$p(y,x)\ \{\ a \leftarrow y,\ x \leftarrow v\ \} = p(y,v)$$

$$p(a, f(x))\ \ \{\ x \leftarrow a,\ f(x) \leftarrow b\ \} = p(a, b)$$

$$p(x, b)\quad \{\ x \leftarrow a,\ f(x) \leftarrow b\ \} = p(a, b)$$

Unification

- Most General Unifier
  - Follow the flow chart on p.20 of lecture #9 notes

Relational Resolution

- Clausal form (INSEADO)
- Negated goal
  - Watch out for quantifiers when negating
- When resolving two clauses, you can
  - Rename the variables
  - Unify the clauses

Relational Resolution

- INSEADO example:

$$\exists y.\forall x.\exists z.[\ (p(x,y)\ \&\ \exists y.q(y)) \Rightarrow (\exists x.\exists y.\exists z.\forall x.p(z,x)\ |\ \forall u.q(x))\ ]$$

Relational Resolution

- Common mistakes:

1. { p(x), q(x) }
2. { ~p(a) }
3. { q(x) }   1,2

1. { ~p(x), q(b) }
2. { p(a), q(x)}
3. No resolvents

1. { p(x), q(x) }
2. { ~p(a) }
3. { p(a), q(a) }   UI
4. { q(a) }   2,3

1. { ~r(x, y), ~r(y, z), r(x, z) }
2. No resolvents

Relational Resolution

- Example:

  "Assume that binary relation R has the properties of irreflexivity and transivity. Using resolution, show that R is asymmetric."

Premises: $\sim r(x,x)$

$r(x,y) \ \& \ r(y,z) \Rightarrow r(x,z)$

Goal: $r(x,y) \Rightarrow \sim r(y,x)$

Relational Resolution

- Example:

$\{ \sim r(x,x) \}$	1. Premise
$\{ r(x,y) \;\&\; r(y,z) \Rightarrow r(x,z) \}$	2. Premise
$\{ r(x,y) \}$	3. Neg Goal
$\{ r(y,x) \}$	4. Neg Goal

Applications

- Logical Entailment

  "is *smart(tim)* logically entailed by the premises?"

  Add: $smart(tim) \Rightarrow goal$

- Answer Extraction / Fill-in-the-Blank Resolution

  "find a term *x* such that *smart(x)* is true"

  Add: $smart(x) \Rightarrow goal(x)$

Strategies

- Elimination Strategies
  - Identical Clause Elimination
  - Pure Literal Elimination
  - Tautology Elimination
  - Subsumption Elimination

- Restriction Strategies
  - Unit Restriction
  - Input Restriction
  - Linear Restriction
  - Set of Support Restriction

Strategies

- All elimination strategies are complete
- Unit Resolution is *incomplete*
  - Lecture 12, Slide 26
- Input Resolution is *incomplete*
  - Lecture 12, Slide 29
- Linear Resolution is *complete*
- Set of Support Resolution is *complete*

Ordered Resolution

- Clause vs. Chain
  - Clause: set of literals

    $\{ p, q, r \}$
  - Chain: sequence of literals

    $< p, q, r >$

- Ordered Resolution
  - "Cancel" the first literals from a pair of chains

- Ordered Resolution is not compatible with many strategies
  - Semi-ordered Resolution
  - Contrapositives
  - Horn Restriction

Ordered Resolution

- Model Elimination
  - Ordered Resolution + Linearity Restriction
  - Works with Set of Support and Input Restrictions
- Reduced Literals
- Reduction, Cancellation, Dropping

Epilog

- Rule form

  < p(x) >

  < q(a), ~r(y,z), s(x) >

- Conclusions as questions

  Prove: p(a) & q(b)

- Backward chaining
  - Similar to reduction in Model Elimination
- Cancellation & Dropping
  - Same as in Model Elimination

Equality

- Two approaches
  - Add axioms of equality
  - Add new rules of inference

Equality

- Add axioms of equality
  - Reflexivity

    $$x = x$$

  - Symmetry

    $$x = y \Leftarrow y = x$$

  - Transitivity

    $$x = z \Leftarrow x = y \wedge y = z$$

  - Flattening (not really an axiom)

    $$f(f(a)) = a \Leftrightarrow \exists x.(f(a) = x \wedge f(x) = a)$$

  - Substitution

    $$f(x) = z \Leftarrow x = y \wedge f(y) = z$$

Equality

- Add rules of inference
  - Demodulation (Lecture 15, Slide 17)
    - Unit equation only
    - Variables substituted in Equation only
  - Paramodulation (Lecture 15, Slide 25)
    - Anything goes?

Equality

- Demodulation examples:

$$\frac{\{\ p(a, f(b, g(a, c)))\ \} \quad \{\ f(x, y) = i(x)\ \}}{\{\ p(a, i(b))\ \}}$$

$$\frac{\{\ p(a, f(x, b))\ \} \quad \{\ f(a, b) = e\ \}}{\{\ p(a, e)\ \}}$$

$$\frac{\{\ p(a, f(b,a))\ ,\ q(b)\ \} \quad \{\ f(x, y) = i(x)\ \}}{\{\ p(a, i(b))\ ,\ q(b)\ \}}$$

$$\frac{\{\ p(a, f(b,a))\ \} \quad \{\ f(x, y) = i(x)\ ,\ q(b)\ \}}{\{\ p(a, i(b))\ ,\ q(b)\ \}}$$

Equality

- Paramodulation examples:

$$\frac{\{\ p(a, f(x, b, c)) ,\ q(x)\ \} \quad \{\ f(a, b, z) = i(z)\ \}}{\{\ p(a, i(c)) ,\ q(a))\ \}}$$

$$\frac{\{\ p(a, f(x, b, c)) ,\ q(x)\} \quad \{\ f(a, b, z) = i(z) ,\ r(z)\ \}}{\{\ p(a, i(c)) ,\ q(b) ,\ r(c)\ \}}$$

Induction

Linear Induction

Input: successor function on individuals

Output: universally quantified conclusions

Structural Induction

Input: constructor function for structures

Output: universally quantified conclusions

Ordered Induction

Input: ordering relation on individuals

Output: universally quantified conclusions

Linear Induction Schema

If a property holds of the base element and if it holds of a successor whenever it holds of an element, it holds of all elements in the universe of discourse.

$$\phi[\mathrm{e}] \wedge \forall x.(\phi[x] \Rightarrow \phi[f(x)]) \Rightarrow \forall x.\phi[x]$$

Base case: $\phi[e]$
Inductive case: $\forall x.(\phi[x] \Rightarrow \phi[f(x)])$
  Inductive antecedent: $\phi[x]$
  Inductive consequent: $\phi[f(x)]$)
Conclusion: $\forall x.\phi[x]$

Linear Induction Method

Using the Induction schema to prove a universally quantified formula.

(1) Base Case. Prove the base case.

(2) Inductive Case. Assume ground version of induction antecedent (induction hypothesis) and prove corresponding version of induction consequent.

If successful, the universally quantified conclusion holds.