Introduction to

Artificial Intelligence

Lecture 9 – Logical reasoning

CS/CNS/EE 154
Andreas Krause

First order logic (FOL)

- Propositional logic is about simple facts
  - "There is a breeze at location [1,2]"

- First order logic is about facts involving
  - *Objects*: Numbers, people, locations, time instants, …
  - *Relations*: Alive, IsNextTo, Before, …
  - *Functions*: MotherOf, BestFriend, SquareRoot, OneMoreThan, …

- Will be able to say:
  - IsBreeze(x); IsPit(x); IsNextTo(x,y)

$$\forall x, y : (IsPit(x) \wedge IsNextTo(x, y)) \Rightarrow IsBreeze(y)$$

FOL: Basic syntactic elements

- Constants: KingJohn, 1, 2, …, [1,1], [1,2], …,[n,n], …
- Variables: x, y, z, …
- Predicates: Brother, >, =, …
- Functions: LeftLegOf, MotherOf, Sqrt, …
- Connectives: $\wedge$, $\vee$, $\neg$
- Quantifiers: $\forall$, $\exists$

- Constant, predicates and functions are mere symbols (i.e., have no meaning on their own)

FOL Syntax: Atomic sentences

A (variable-free) term is a

- constant symbol or
- k-ary function symbol: *function(term$_1$,term$_2$, …, term$_k$)*

Example: *LeftLegOf(KingJohn)*

Richard

Succ (Succ (… (Succ (Zero))))

An atomic sentence is a predicate symbol applied to terms

Example:

- *Brother(KingJohn, RichardLionheart)*
- *IsNextTo([1,1],[1,2])*
- *> (Length(LeftLegOf(KingJohn)), Length(LeftLegOf (RichardLionheart)))*

Models in FOL

- Much more complicated than in Propositional Logic
- Models contain
  - Set of objects (finite or countable)
  - Set of relations between objects (map obj's to truth values)
  - Set of functions (map objects to other objects)

  and their interpretations:
  - Mapping from constant symbols to model objects
  - Mapping from predicate symbols to model relations
  - Mapping from function symbols to model functions
- An atomic sentence $predicate(term_1, term_2, \ldots, term_k)$ is true if the objects referred to by $term_1, term_2, \ldots, term_k$ are in the relation referred to by *predicate*

Quantifiers

- Allow variables in addition to constants

$$Homework(x, 154)$$

- Sentences with free variables: $S(x)$
- Quantifiers bind free variables

$\forall x : S(x)$ is true if S(x) is true for all instantiations of x (i.e., for each possible object in the model)

$\exists x : S(x)$ is true if S(x) is true for at least one instantiation of x (i.e., for some object)

- Example:
  - All homeworks in 154 are hard

    $\forall x : (Homework(x, 154) \Rightarrow Hard(x))$

  - At least one of the 154 homeworks is hard

    $\exists x : Homework(x, 154) \wedge Hard(x)$

$$\forall x : Homework(x, 154) \Rightarrow Hard(x)$$

Properties of quantifiers

- Is $\forall x \;\; \forall y \;\; S(x,y)$ the same as $\forall y \;\; \forall x \;\; S(x,y)$?

- Is $\exists x \;\; \exists y \;\; S(x,y)$ the same as $\exists y \;\; \exists x \;\; S(x,y)$?

- Is $\exists x \;\; \forall y \;\; S(x,y)$ the same as $\forall y \;\; \exists x \;\; S(x,y)$?

$\exists x \, \forall y \; Loves(x,y)$ There is someone who loves everyone

$\forall y \, \exists x \; Loves(x,y)$ Everybody is loved by someone

Examples

- Brothers are siblings

$$\forall x,y \; (Brothers(x,y) \Rightarrow Siblings(x,y))$$

- "Siblings" is symmetric

$$\forall x,y \; (Siblings(x,y) \Rightarrow Siblings(y,x))$$

- A mother is somebody's female parent

$$\forall x \quad IsMother(x) \Leftrightarrow [\exists y \, Female(x) \wedge Parent(x,y)]$$

- A first cousin is a child of a parent's sibling

$$\forall x,y \; FirstCousin(x,y) \Leftrightarrow \exists w,z \, (Parent(w,x) \wedge Parent(z,y) \wedge Sibling(w,z))$$

Equality: Special predicate

- Reflexive, transitive and symmetric $E(x,y)$, abbrev. $x=y$

  $\forall x: \ x=x$

  $\forall x,y,z: \ x=y \wedge y=z \Rightarrow x=z$

  $\forall x,y: \ x=y \Rightarrow y=x$

- Substituting equal objects doesn't change value of expressions $S(x)$

  $\forall x,y: \ x=y \Rightarrow (S(x) \Leftrightarrow S(y))$

- All models need to satisfy these properties
  - Typically, just assume that model has an "equality" relation, and the interpretation of the "=" symbol refers to that relation

Wumpus world in FOL

- Modeling perception

$$\forall b, g, t \;\; Percept([Smell, b, g], t) \Rightarrow Smelt(t)$$
$$\forall s, g, t \;\; Percept([s, Breeze, g], t) \Rightarrow Breezy(t)$$

- Properties of locations

$$\forall x, t \;\; At(Agent, x, t) \land Smelt(t) \Rightarrow Smelly(x)$$
$$\forall x, t \;\; At(Agent, x, t) \land Breeze(t) \Rightarrow Breezy(x)$$

- Squares are breezy near a pit

$$\forall y \;\; Breezy(y) \Leftrightarrow \Big(\exists x \;\; Pit(x) \land Adjacent(x, y)\Big)$$

Modeling change

- Facts hold only in certain situations

  $Holding(Gold, t)$ instead of $Holding(Gold)$

- Can model using situation calculus
  - Add situation argument to each non-eternal predicate
  - E.g., *t* in $Holding(Gold, t)$

- Model effects of actions using *Result* function

  Result(a,s) is the situation that results from doing *a* in *s*

- Effect axioms model changes due to actions

$$\forall s \;\; AtGold(s) \Rightarrow Holding(Gold, Result(Grab, s))$$

The frame problem

- In addition to modeling change, also need to model non-change
- Need "frame" axioms that model non-change

$$\forall s \;\; HaveArrow(s) \Rightarrow HaveArrow(Result(Grab, s))$$

- The frame problem:
  - Number of frame axioms can be large
  - Causes problems in inference

Solving the frame problem

- Successor state axioms:

  For each *non-eternal* predicate, model how it is *affected or not affected* by actions

- P is true ⇔ [an action made P true or P is already true and no action made it false]

- Example for holding the gold

$$\forall a,s \;\; Holding(Gold, Result(s,a)) \Leftrightarrow [(AtGold(s) \wedge (a = Grab))$$
$$\vee (Holding(Gold, s) \wedge \neg(a = Release))]$$

Planning using FOL

- Knowledge base (KB) contains all known facts
  - Successor state axioms
  - Properties of locations / perception
  - Initial conditions: $At(Agent, [1,1], S_0)$

$$Percept([NoSmell, Breeze, NoGlitter], S_0)$$

- Use inference to find whether, e.g.,
  - Agent can obtain gold: $KB \models \exists s \;\; Holding(Gold, s)$

    Example response:

$$\{s/Result(Grab, Result(Forward, S_0))\}$$

  - Agent can safely move to [2,2]

$$KB \models \exists s \;\; At(Agent, [2,2], s) \wedge Alive(s)$$

Inference in FOL

- Much more complicated then in propositional logic

- Two approaches
  - Propositionalization: Convert to propositional formula and use propositional inference

  - “Lifted” inference: Syntactically manipulate propositional sentences directly

Propositionalization

- Create a propositional symbol for each atomic sentence

$\underbrace{Breezy([1,1])}_{C_0}, \underbrace{Adjacent([1,1],[1,2])}_{C_1}$

- Inferring propositional sentences by grounding universally quantified variables

$\forall x : Breezy(x) \Rightarrow Breezy([1,1]) \wedge Breezy([1,2]) \wedge \ldots$

- Replace existential quantifier by introducing new constants

$\exists x : Breezy(x) \Rightarrow C'$

Problems with propositionalization

- May need to create lots of unnecessary symbols
- More importantly: Number of symbols could be *infinite*
  - If KB involves functions, can build infinitely many terms

Odd(x) ; Zero , Succ(·) = Odd(Zero) , Odd(Succ(Zero)) , Odd(Succ(Succ(

Theorem (Herbrand '30)

If a sentence α is entailed by a FOL KB, then there is a proof using only a finite subset of the propositionalized KB

Naïve algorithm for FOL using propositionalization

- Want to determine whether sentence α is entailed by KB

$i \leq j \Rightarrow |PKB_i| \leq |PKB_j|$

- Can enumerate all finite subsets PKB_1, PKB_2, …
  of the propositionalized knowledge base $KB \wedge \neg\alpha$

For i = 1 to ∞

  If PKB_i is unsatisfiable, e.g., using propositional resolution:

    break and return *true*

- If $KB \models \alpha$ above algorithm stops after a finite number of steps
- If $KB \not\models \alpha$ it will never stop
- This is intrinsic: FOL is semi-decidable

Lifted inference

- Want to operate on FOL sentences directly
- Suppose we know

$$\forall x \;\; King(x) \wedge Greedy(x) \Rightarrow Evil(x)$$

p_1' p_2'

$$King(John) \;\; \text{and} \;\; Greedy(John)$$

p_1 p_2

$\theta = \{ x / John \}$ $\quad KB \vdash_{GMP} Evil(John)$

- Lifted inference allows to infer $Evil(John)$ without instantiating

$$King(John) \wedge Greedy(John) \Rightarrow Evil(John)$$

$\wedge \; King(Richard) \wedge Greedy(Richard) \Rightarrow Evil(Richard)$

$\wedge \; \ldots$

$\wedge \; \ldots$

Generalized modus ponens

- Let $p'_1, \ldots, p'_n$ and $p_1, \ldots, p_n$ be FOL sentences
- Suppose $\mathrm{Subst}(\theta, p'_i) = \mathrm{Subst}(\theta, p_i)$
  where θ is a substitution and $\mathrm{Subst}(\theta, p)$ its application
- E.g.: $\theta = \{x / John\}$

  $p_i' = King(x)$ $\quad$ $p_i = King(Richard)$

  Then $Subst(\theta, p_i') = King(John)$ $\quad$ $Subst(\theta, p_i) = King(Richard)$

  Then the following inference is sound:

$$\frac{p'_1, \ldots, p'_n, \quad p_1 \wedge \cdots \wedge p_n \Rightarrow q}{\mathrm{Subst}(\theta, q)}$$

  Can use in a generalization of forward/backward chaining

  However, GMP is not complete

Generalization resolution

- Can also develop a lifted variant of resolution
- Details in reading
- Generalized resolution is sound and refutation-complete

➔ Can prove $KB \vdash_{GRes} \alpha$ if $KB \models \alpha$

➔ Cannot prove $KB \nvdash_{GRes} \alpha$ if $KB \not\models \alpha$

Other logics

Language	Ontological commitment	Epistemological commitment
Propositional logic	facts	true/false/unknown
First order logic	facts, objects, relations	true/false/unknown
Higher order logic	facts, objects, relations, relations of relations, …	true/false/unknown
Temporal logic	facts, objects, relations, times	true/false/unknown
Fuzzy logic	facts	degree of truth in [0,1]
Bayesian networks (up next)	facts	belief in probability of truth
Bayesian logic / Markov Logic Networks	facts, objects, relations	belief in probability of truth