Knowledge Representation IV
Inference for First-Order Logic

CSE 473

FOL Reasoning

- Basics of FOL reasoning
- Classes of FOL reasoning methods
  - Forward & Backward Chaining
  - Resolution
  - Compilation to SAT

Basics: Universal Instantiation

- Universally quantified sentence:
  - $\forall$x: Monkey(x) ^ Curious(x) → Fuzzy(x)
- Intutively, x can be anything:
  - Monkey(George) ^ Curious(George) → Fuzzy(George)
  - Monkey(Peter) ^ Curious(Peter) → Fuzzy(Peter)
  - Monkey(DadOf(George)) ^ Curious(DadOf(George)) → Fuzzy(DadOf(George))
- Formally: (example)

$$\frac{\forall x \;\; S}{\mathbf{Subst(\{x/p\},\ S)}} \qquad \frac{\forall x \;\; \mathbf{Monkey(x) \rightarrow Curious(x)}}{\mathbf{Monkey(George) \rightarrow Curious(George)}}$$

x is replaced with p in S, and the quantifier removed

x is replaced with George in S, and the quantifier removed

Basics: Existential Instantiation

- Existentially quantified sentence:
  - $\exists$x: Monkey(x) ^ ¬Curious(x)
- **Intutively, x must name something. But what?**
  - ??? Monkey(George) ^ ¬Curious(George) ???
  - No! S might not be true for George!
- Use a *Skolem Constant* :
  - Monkey(k) ^ ¬Curious(k)
  - …where k is a **completely new** symbol
- Formally: (example)

$$\frac{\exists x \;\; S}{\mathbf{Subst(\{x/k\},\ S)}} \qquad \frac{\exists x \;\; \mathbf{Monkey(x) \rightarrow Curious(x)}}{\mathbf{Monkey(newGuy) \rightarrow Curious(newGuy)}}$$

newGuy is the Skolem constant

Basics: Generalized Skolemization

- What if our existential variable is nested?

  $\forall x\ \exists y$: Monkey(x) → HasTail(x, y)

  ??? $\forall x$: Monkey(x) → HasTail(x, skolemTail) ???

- Existential variables can be replaced by

  Skolem functions (or constants)

  Args to function are all surrounding $\forall$ vars

- $\forall d\ \exists t$ has(d, t)

  $\forall d$ has(d, f(d))

- $\exists x\ \forall y$ *loves(y, x)*

  $\forall y$ loves(y, f())

  $\forall y$ loves(y, f_{97})

Basics: Unification

- What if we want to use modus ponens?

  a ^ b → c

  a ^ b

  c

  Fuzzy(x) ^ Monkey(x) → Curious(x)

  Fuzzy(George) ^ Monkey(George)

  ????

- Must ***unify*** our expressions

Unification

- Match up expressions by finding variable values that make the expressions identical

  Variables denoted *?x*

- Unify(x, y) returns "mgu"

  Unify(city(?a), city(kent)) returns {?a/kent}

- Substitute(expr, mapping) returns new expr

  Substitute(connected(?a, ?b), {?a/kent})

  returns connected(kent, ?b)

Unification Examples I

- Unify(road(?a, kent), road(seattle, ?b))

  Unification ok

  Returns {?a / seattle, ?b / kent}

  When substituted in both expressions, they match.

  Each is **(road(seattle, kent))**

- Unify(road(?a, ?a), road(seattle, kent))

  Impossible: ?a can't be seattle and kent at the same time!

Unification Examples II

- Unify(f(g(?x, dog), ?y)), f(g(cat, ?y), dog)
  {?x / cat, ?y / dog}
- Unify(f(g(?x)), f(?x))
  They don't unify: no substitution makes them the same.
  E.g. consider: {?x / g(?x) }
  We get f(g(g(?x))) and f(g(?x)) … not equal!
- Thus: A variable value may not ***contain*** itself
  Directly or indirectly

Unification Examples III

- Unify(f(g(cat, dog), ?y)), f(?x), dog)
  {?x / g(cat, dog), ?y / dog}
- Unify(f(g(?y)), f(?x))
  {?x / g(?y), ?y / ?y}
- Back to fuzzy monkeys:

$$\frac{\begin{array}{l}\text{Fuzzy(x) ^ Monkey(x)} \rightarrow \text{Curious(x)} \\ \text{Fuzzy(George) ^ Monkey(George)}\end{array}}{\text{Curious(George)}}$$

Unify and then use modus ponens = *generalized modus ponens*

Inference I: Forward Chaining

- Given:
  ∀x Monkey(x) ^ Fuzzy(x) → Curious(x)
  ∀y Fuzzy(y)
  Monkey(George)

  Prove:
  Curious(George)
- The algorithm:
  Start with the KB
  Add any fact you can generate with GMP
  Repeat until: goal reached or generation halts.
- Sound? Complete? Decidable?
- Speed concerns?
  Unification; premise rechecking; irrelevant fact gen.

Inference II: Backward Chaining

- Given:
  ∀x Monkey(x) ^ Fuzzy(x) → Curious(x)
  ∀y Fuzzy(y)
  Monkey(George)

  Prove:
  Curious(George)
- The algorithm:
  Start with KB and goal.
  Find all rules whose *results* unify with the goal:
    Add the *bodies* of these rules to the goal list
    Remove the corresponding result from the goal list
  Stop when:
    Goal list is empty (SUCCEED)
    Progress halts (FAIL)

Inference III: Resolution

[Robinson 1965]

$\{ (p \vee \alpha), (\neg p \vee \beta \vee \gamma) \} \ \vdash_R \ \underbrace{(\alpha \vee \beta \vee \gamma)}$

Recall Propositional Case:

- **Literal in one clause**
- **Its negation in the other**
- **Result is disjunction of *other* literals**

First-Order Resolution

[Robinson 1965]

$\{ (p(?x) \vee a(a), \quad (\neg p(q) \vee b(?x) \vee c(?y)) \}$

$\vdash_R$

$(a(a) \vee b(q) \vee c(?y))$

- **Literal in one clause**
- **The negation of *something which unifies* in the other**
- **Result is disjunction of other literals / mgu**

First-Order Resolution

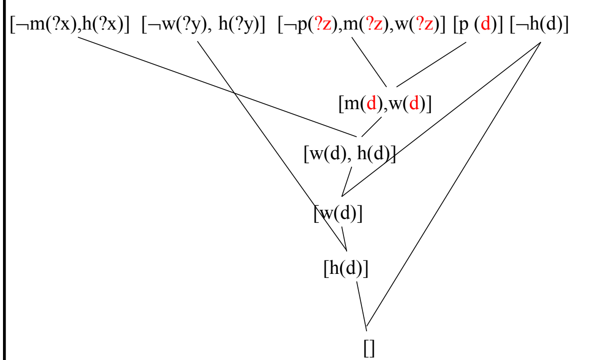
- Answers: Is it the case that $\Sigma \models \Phi$?
- Method
  - Let S = KB $\wedge$ ¬goal
  - Convert S to clausal form
    - Standardize variables
    - Move quantifiers to front, skolemize to remove ∃
    - Replace ⇒ with ∨ and ¬
    - Demorgan's laws to get CNF (ands-of-ors)
  - Resolve S goal until get empty clause

First-Order Resolution Example

- Given
  - ∀?x man(?x) => human(?x)
  - ∀?x woman(?x) => human(?x)
  - ∀?x prof(?x) => man(?x) ∨ woman(?x)
  - prof(dieter)
- Prove
  - human(dieter)

[¬m(?x),h(?x)] [¬w(?y), h(?y)] [¬p(?z),m(?z),w(?z)] [p(d)][¬h(d)]

Example Continued



Resolution Example 2

Given

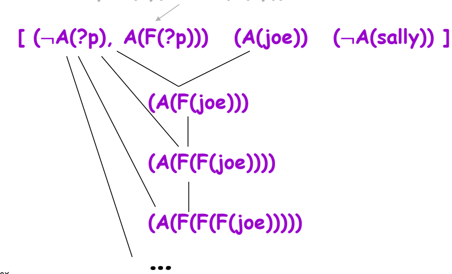
∀?p ∃?f A(?p) => A(?f)

A(joe)

∀?p A(?p) => A(F(?p))

Prove

A(sally)



Inference IV: Compilation to Prop. Logic

- Sentence S:
  $\forall_{city}$ *a,b* connected(a,b)
- Universe
  Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula:

Cst ∧ Cse ∧ Cts ∧ Cte ∧ Ces ∧ Cet

Compilation to Prop. Logic (cont)

- Sentence S:
  $\exists_{city}$ *c* biggest(c)
- Universe
  Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula:

Bs ∨ Bt ∨ Be

Compilation to Prop. Logic (cont again)

- Universe
  - Cities: seattle, tacoma, enumclaw
  - Firms: IBM, Microsoft, Boeing
- First-Order formula

  $\forall_{firm} f \ \exists_{city} c \quad HeadQuarters(f, c)$
- Equivalent propositional formula

$$[\ (HQis \vee HQit \vee HQie) \wedge (HQms \vee HQmt \vee HQme) \wedge (HQbs \vee HQbt \vee HQbe)\ ]$$

Hey!

- You said FO Inference is semi-decidable
- But you compiled it to SAT

  Which is NP Complete
- So now we can always do the inference?!?

  Tho it might take exponential time…
- Something seems wrong here….????

Compilation to Prop. Logic (cont for the last time)

- Universe
  - People: homer, bart, marge
- First-Order formula

  $\forall_{people}\, p \quad Male(FatherOf(p))$
- Equivalent propositional formula

$$[\ (M_{father\text{-}homer} \wedge M_{father\text{-}bart} \wedge M_{father\text{-}marge} \wedge (M_{father\text{-}father\text{-}homer} \wedge M_{father\text{-}father\text{-}bart} \wedge \cdots (M_{father\text{-}father\text{-}father\text{-}homer} \wedge \cdots \ldots]$$

Restricted Forms of FO Logic

- Known, Finite Universes

  Compile to SAT
- Frame Systems

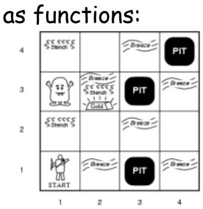
  Ban certain types of expressions
- Horn Clauses (at most one negative literal)

  Aka Prolog
- Function-Free Horn Clauses

  Aka Datalog

Back To the Wumpus World

- Recall description:
  - Squares as lists: [1,1] [3,4] etc.
  - Square adjacency as binary predicate.
  - Pits, breezes, stenches as unary predicates: Pit(x)
  - Wumpus, gold, homes as functions: WumpusHome(x)



Back To the Wumpus World

- "Squares next to pits are breezy":

  ∀x, y, a, b:

  Pit([x, y]) ^ Adjacent([x, y], [a, b]) → Breezy([a, b])

- "Breezes happen *only* and *always* next to pits"

  ∀a,b Breezy([a, b]) <=>

  ∃ x,y Pit([x, y]) ^ Adjacent([x, y], [a, b])

Back To the Wumpus World

- Given:

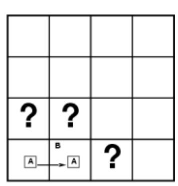
  ∀a,b Breezy([a, b]) <=>

  ∃ x,y Pit([x, y]) ^ Adjacent([x, y], [a, b])

  Breezy([1,2])

- Prove:

  Pit([3,2]) v Pit([2,2])



What About Our Agent?

- Still don't know how to deal with time
- Still don't know how to go from knowledge of the world to *action* in the world

  → Planning