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:
  \( \forall x: \text{Monkey}(x) \land \text{Curious}(x) \rightarrow \text{Fuzzy}(x) \)
- Intuitively, \( x \) can be anything:
  \( \text{Monkey}(\text{George}) \land \text{Curious}(\text{George}) \rightarrow \text{Fuzzy}(\text{George}) \)
  \( \text{Monkey}(\text{Peter}) \land \text{Curious}(\text{Peter}) \rightarrow \text{Fuzzy}(\text{Peter}) \)
  \( \text{Monkey}(\text{DadOf}(\text{George})) \land \text{Curious}(\text{DadOf}(\text{George})) \rightarrow \text{Fuzzy}(\text{DadOf}(\text{George})) \)
- Formally:
  \( \forall x: S \)
  \( \text{Subst\{x/p\}, S} \)
  \( \forall x: \text{Monkey}(x) \rightarrow \text{Curious}(x) \)

Basics: Existential Instantiation

- Existentially quantified sentence:
  \( \exists x: \text{Monkey}(x) \land \neg\text{Curious}(x) \)
- Intuitively, \( x \) must name something. But what?
  \( ??? \; \text{Monkey}(\text{George}) \land \neg\text{Curious}(\text{George}) \; ??? \)
  No! \( S \) might not be true for \( \text{George} \)!
- Use a Skolem Constant:
  \( \text{Monkey}(k) \land \neg\text{Curious}(k) \)
  ...where \( k \) is a completely new symbol
- Formally:
  \( \exists x: S \)
  \( \text{Subst\{x/k\}, S} \)
  \( \exists x: \text{Monkey}(x) \rightarrow \text{Curious}(x) \)
  \( \text{newGuy} \) is the Skolem constant
Basics: Generalized Skolemization

- What if our existential variable is nested?
  \( \forall x \exists y: \text{Monkey}(x) \rightarrow \text{HasTail}(x, y) \)
  \( ??? \forall x: \text{Monkey}(x) \rightarrow \text{HasTail}(x, \text{skolemTail}) ??? \)

- Existential variables can be replaced by
  Skolem functions (or constants)
  Args to function are all surrounding \( \forall \) vars

  \( \forall d \exists t \ \text{has}(d, t) \)
  \( \forall d \ \text{has}(d, f(d)) \)

  \( \exists x \forall y \ \text{loves}(y, x) \)
  \( \forall y \ \text{loves}(y, f()) \)
  \( \forall y \ \text{loves}(y, f_{97}) \)

Basics: Unification

- What if we want to use modus ponens?
  \( a \land b \rightarrow c \)
  \( a \land b \)
  \( c \)

  \( \text{Fuzzy}(x) \land \text{Monkey}(x) \rightarrow \text{Curious}(x) \)
  \( \text{Fuzzy}(\text{George}) \land \text{Monkey}(\text{George}) \)
  ???

- Must **unify** our expressions

Unification

- Match up expressions by finding variable values that make the expressions identical
  Variables denoted \(?x\)

- \( \text{Unify}(x, y) \) returns "mgu"
  \( \text{Unify}(\text{city}(?a), \text{city}(\text{kent})) \) returns \(?a/\text{kent}\)

- \( \text{Substitute}(\text{expr}, \text{mapping}) \) returns new expr
  \( \text{Substitute}(\text{connected}(?a, ?b), (?a/\text{kent})) \)
  returns \(\text{connected}(\text{kent}, ?b)\)

Unification Examples I

- \( \text{Unify}(\text{road}(?a, \text{kent}), \text{road}(\text{seattle}, ?b)) \)
  Unification ok
  Returns \(?a/\text{seattle}, \ ?b/\text{kent}\)
  When substituted in both expressions, they match.
  Each is \(\text{road}(\text{seattle}, \text{kent})\)

- \( \text{Unify}(\text{road}(?a, ?a), \text{road}(\text{seattle}, \text{kent})) \)
  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)
  \( \{?x / \text{cat}, \ ?y / \text{dog}\} \)
- 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)
  \( \{?x / g(\text{cat, dog}), \ ?y / \text{dog}\} \)
- Unify(f(g(?y)), f(?x))
  \( \{?x / g(?y), \ ?y / ?y\} \)
- Back to fuzzy monkeys:
  \[ \text{Fuzzy}(x) \land \text{Monkey}(x) \rightarrow \text{Curious}(x) \]
  \[ \text{Fuzzy}(\text{George}) \land \text{Monkey}(\text{George}) \rightarrow \text{Curious}(\text{George}) \]
  Unify and then use modus ponens = generalized modus ponens

Inference I: Forward Chaining

- Given:
  \[ \forall x \: \text{Monkey}(x) \land \text{Fuzzy}(x) \rightarrow \text{Curious}(x) \]
  \[ \forall y \: \text{Fuzzy}(y) \]
  \[ \text{Monkey}(\text{George}) \]

- 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:
  \[ \forall x \: \text{Monkey}(x) \land \text{Fuzzy}(x) \rightarrow \text{Curious}(x) \]
  \[ \forall y \: \text{Fuzzy}(y) \]
  \[ \text{Monkey}(\text{George}) \]

- 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)
**First-Order Resolution**

- **Answers**: Is it the case that $\Sigma \models \Phi$?
- **Method**
  - Let $S = KB \land \neg \text{goal}$
  - Convert $S$ to clausal form
    - Standardize variables
    - Move quantifiers to front, skolemize to remove $\exists$
    - Replace $\Rightarrow$ with $\lor$ and $\land$
    - Demorgan’s laws to get CNF (ands-of-ors)
  - Resolve $S$ goal until get empty clause

**First-Order Resolution Example**

- **Given**
  
  $\forall x \text{ man}(?x) \Rightarrow \text{human}(?x)$
  $\forall x \text{ woman}(?x) \Rightarrow \text{human}(?x)$
  $\forall x \text{ prof}(?x) \Rightarrow \text{man}(?x) \lor \text{woman}(?x)$
  $\text{prof}(\text{dieter})$

- **Prove**
  
  $\text{human}(\text{dieter})$

\[-m(?x),h(?x)] \ [\neg w(?y), \ h(?y)] \ [\neg p(?z), m(?z), w(?z)] \ [p(d)\neg h(d)]
Example Continued

\[
\begin{align*}
[m(?x), h(?x)] & \quad [-w(?y), h(?y)] \\
[-p(?z), m(?z), w(?z)] & \quad [p \ (d)] \quad [-h(d)] \\
\end{align*}
\]

Resolution Example 2

Given

\[
\forall ?p \exists ?f \quad A(?p) \Rightarrow A(?f) \\
A(\text{joe})
\]

Prove

\[
\forall ?p \ A(?p) \Rightarrow A(F(?p)) \\
\begin{array}{c}
(\neg A(?p), A(F(?p))) \\
A(\text{joe}) \\
A(sally)
\end{array}
\]

Inference IV:
Compilation to Prop. Logic

- Sentence S:
  \[
  \forall \text{city} \ a, b \ \text{connected}(a, b)
  \]
- Universe
  Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula:
  \[
  Cst \land Cse \land Cts \land Cte \land Ces \land Cet
  \]

Compilation to Prop. Logic (cont)

- Sentence S:
  \[
  \exists \text{city} \ c \ \text{biggest}(c)
  \]
- Universe
  Cities: seattle, tacoma, enumclaw
- Equivalent propositional formula:
  \[
  Bs \lor Bt \lor Be
  \]
Compilation to Prop. Logic
(cont again)

- Universe
  - Cities: seattle, tacoma, enumclaw
  - Firms: IBM, Microsoft, Boeing
- First-Order formula
  \[ \forall_{\text{firm } f} \exists_{\text{city } c} \text{HeadQuarters}(f, c) \]
- Equivalent propositional formula
  \[ \left( \text{HQis} \lor \text{HQit} \lor \text{HQie} \right) \land \]
  \[ \left( \text{HQms} \lor \text{HQmt} \lor \text{HQme} \right) \land \]
  \[ \left( \text{HQbs} \lor \text{HQbt} \lor \text{HQbe} \right) \]

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
  \[ \forall_{\text{people } p} \text{Male(FatherOf}(p)) \]
- Equivalent propositional formula
  \[ \left( \text{Mfather-homer} \land \text{Mfather-bart} \land \text{Mfather-marge} \right) \land \]
  \[ \left( \text{Mfather} - \text{father-homer} \land \text{Mfather} - \text{father-bart} \land \ldots \right) \]
  \[ \left( \text{Mfather} - \text{father-homer} \land \ldots \right) \]

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":
  \( \forall x, y, a, b: \) 
  \( \text{Pit}([x, y]) \land \text{Adjacent}([x, y], [a, b]) \rightarrow \text{Breezy}([a, b]) \)

• "Breezes happen only and always next to pits":
  \( \forall a, b \) 
  \( \text{Breezy}([a, b]) \iff \exists x, y \text{Pit}([x, y]) \land \text{Adjacent}([x, y], [a, b]) \)

Back To the Wumpus World

• Given:
  \( \forall a, b \) 
  \( \text{Breezy}([a, b]) \iff \exists x, y \text{Pit}([x, y]) \land \text{Adjacent}([x, y], [a, b]) \)
  \( \text{Breezy}([1,2]) \)

• Prove:
  \( \text{Pit}([3,2]) \lor \text{Pit}([2,2]) \)

Back To the Wumpus World

• 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
    \( \rightarrow \) Planning