Inference in First Order Logic

Inference rules

- Universal elimination:
  - $\forall$ x Likes(x, IceCream) with the substitution {x / Einstein} gives us Likes(Einstein, IceCream)
  - The substitution has to be done by a ground term
- Existential elimination:
  - From $\exists$ x Likes(x, IceCream) we may infer Likes(Man, IceCream) as long as Man does not appear elsewhere in the Knowledge base
- Existential introduction:
  - From Likes(Monalisa, IceCream) we can infer $\exists$ x Likes(x, IceCream)

Reasoning in first-order logic

- The law says that it is a crime for a Gaul to sell potion formulas to hostile nations.
- The country Rome, an enemy of Gaul, has acquired some potion formulas, and all of its formulas were sold to it by Druid Traitorix.
- Traitorix is a Gaul.
- Is Traitorix a criminal?

Generalized Modus Ponens

- For atomic sentences p_i, p_i', and q, where there is a substitution θ such that SUBST(θ, p_i') = SUBST(θ, p_i), for all i:

$$\frac{p_1', p_2', \ldots, p_n', (p_1 \wedge p_2 \wedge \ldots \wedge p_n \Rightarrow q)}{SUBST(\theta, q)}$$

Unification

UNIFY(p,q) = θ where SUBST(θ,p) = SUBST(θ,q)

Examples:

UNIFY(Knows(Erdos, x),Knows(Erdos, Godel))
= {x / Godel}

UNIFY(Knows(Erdos, x), Knows(y,Godel))
= {x/Godel, y/Erdos}

Unification

UNIFY(p,q) = θ where SUBST(θ,p) = SUBST(θ,q)

Examples:

UNIFY(Knows(Erdos, x), Knows(y, Father(y)))
= { y/Erdos, x/Father(Erdos) }

UNIFY(Knows(Erdos, x), Knows(x, Godel)) = F

We require the most general unifier

Reasoning with Horn Logic

- We can convert Horn sentences to a canonical form and then use generalized Modus Ponens with unification.
  - We skolemize existential formulas and remove the universal ones
  - This gives us a conjunction of clauses, that are inserted in the KB
  - Modus Ponens help us in inferring new clauses
- Forward and backward chaining

Completeness issues

- Reasoning with Modus Ponens is incomplete
- Consider the example –

$$\forall x\ P(x) \Rightarrow Q(x) \qquad \forall x\ \neg P(x) \Rightarrow R(x)$$
$$\forall x\ Q(x) \Rightarrow S(x) \qquad \forall x\ R(x) \Rightarrow S(x)$$

- We should be able to conclude $S(A)$
- The problem is that $\forall x\ \neg P(x) \Rightarrow R(x)$ cannot be converted to Horn form, and thus cannot be used by Modus Ponens

Godel’s Completeness Theorem

- For first-order logic, any sentence that is entailed by another set of sentences can be proved from that set
  - Godel did not suggest a proof procedure
  - In 1965 Robinson published his resolution algorithm

- **Entailment in first-order logic is semi-decidable, that is, we can show that sentences follow from premises if they do, but we cannot always show if they do not.**

The validity problem of first-order logic

- [Church] The validity problem of the first-order predicate calculus is partially solvable.
- Consider the following formula:

$$[\bigwedge_{i=1}^{n} p(f_i(a), g_i(a))$$

$$\wedge \forall x \forall y [p(x, y) \Rightarrow \bigwedge_{i=1}^{n} p(f_i(x), g_i(x))]]$$

$$\Rightarrow \exists z\, p(z, z)$$

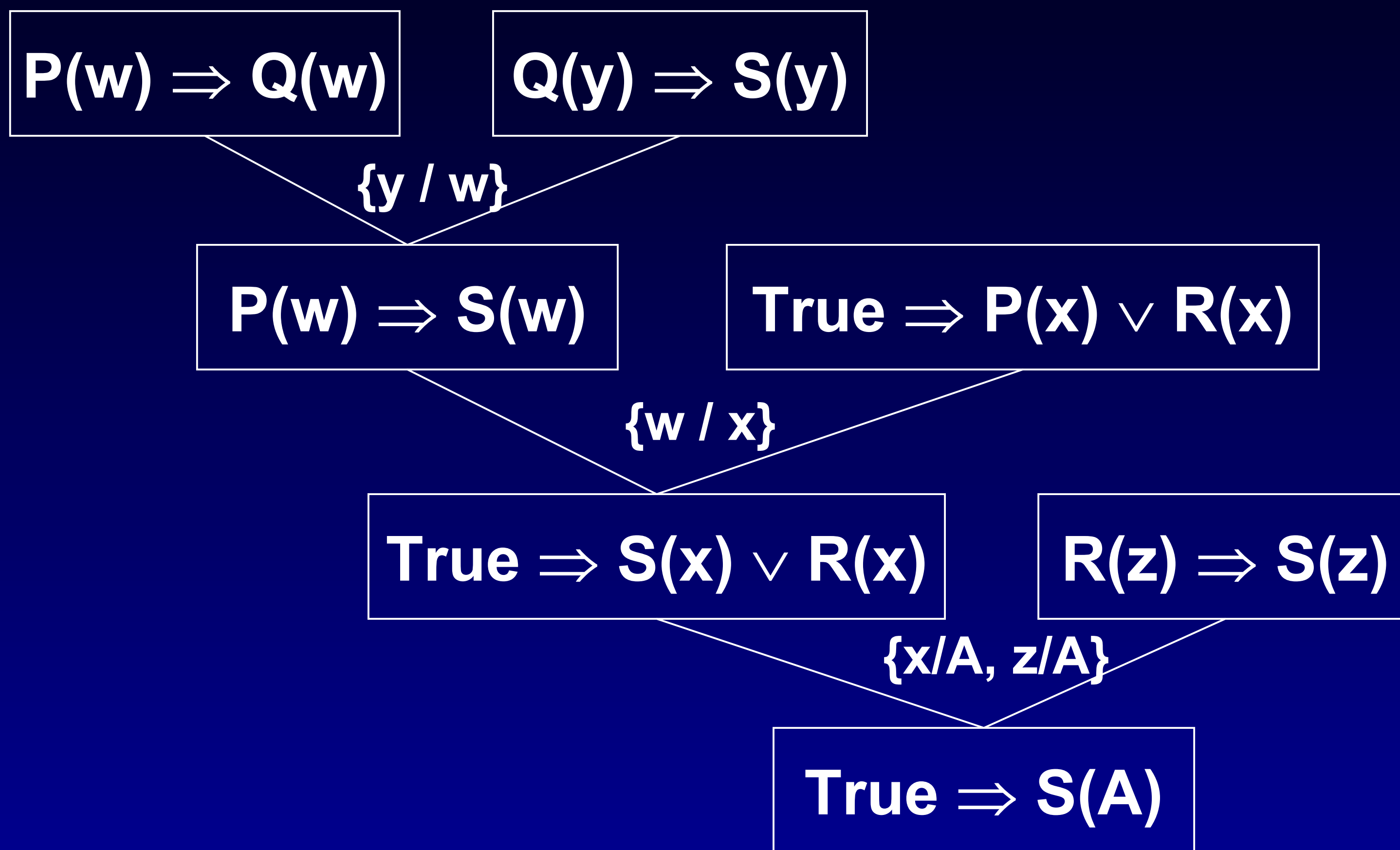
Resolution

- Generalized Resolution Rule:

  For atoms p_i, q_i, r_i, s_i, where Unify(p_j, q_k) = θ, we have:

$$\frac{\begin{array}{c} p_1 \wedge \ldots p_j \ldots \wedge p_{n1} \Rightarrow r_1 \vee \ldots r_{n2} \\ s_1 \wedge \ldots \wedge s_{n3} \Rightarrow q_1 \vee \ldots q_k \ldots \vee q_{n4} \end{array}}{\begin{array}{l} \text{SUBST}(\theta, \\ \quad p_1 \wedge \ldots p_{j-1} \wedge p_{j+1} \ldots \wedge p_{n1} \wedge s_1 \wedge \ldots s_{n3} \\ \qquad \Rightarrow r_1 \vee \ldots r_{n2} \vee \ldots q_{k-1} \vee q_{k+1} \vee \ldots \vee q_{n4}) \end{array}}$$

Our earlier example



Conversion to Normal Form

- A formula is said to be in clause form if it is of the form:

$$\forall x_1 \ \forall x_2 \ \ldots \ \forall x_n \ [C_1 \land C_2 \land \ldots \land C_k]$$

- All first-order logic formulas can be converted to clause form
- We shall demonstrate the conversion on the formula:

$$\forall x \ \{p(x) \Rightarrow \exists z \ \{ \neg \forall y \ [q(x,y) \Rightarrow p(f(x_1))] \land \forall y \ [q(x,y) \Rightarrow p(x)] \}\}$$

Conversion to Normal Form

- Step1: *Take the existential closure and eliminate redundant quantifiers.* This introduces $\exists x_1$ and eliminates $\exists z$, so:

$$\forall x \{p(x) \Rightarrow \exists z \{ \neg\forall y [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall y [q(x,y) \Rightarrow p(x)] \}\}$$

$$\exists x_1 \forall x \{p(x) \Rightarrow \{ \neg\forall y [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall y [q(x,y) \Rightarrow p(x)] \}\}$$

Conversion to Normal Form

- Step 2: *Rename any variable that is quantified more than once.* y has been quantified twice, so:

$$\exists x_1\ \forall x\ \{p(x) \Rightarrow \{\ \neg\forall y\ [q(x,y) \Rightarrow p(f(x_1))]\ \wedge\ \forall y\ [q(x,y) \Rightarrow p(x)]\ \}\}$$

$$\exists x_1\ \forall x\ \{p(x) \Rightarrow \{\ \neg\forall y\ [q(x,y) \Rightarrow p(f(x_1))]\ \wedge\ \forall z\ [q(x,z) \Rightarrow p(x)]\ \}\}$$

Conversion to Normal Form

- Step 3: *Eliminate implication.*

$$\exists x_1 \ \forall x \ \{p(x) \Rightarrow \{ \neg \forall y \ [q(x,y) \Rightarrow p(f(x_1))] \wedge \forall z \ [q(x,z) \Rightarrow p(x)] \}\}$$

$$\exists x_1 \ \forall x \ \{\neg p(x) \vee \{ \neg \forall y \ [\neg q(x,y) \vee p(f(x_1))] \wedge \forall z \ [\neg q(x,z) \vee p(x)] \}\}$$

Conversion to Normal Form

- Step 4: *Move $\neg$ all the way inwards.*

$$\exists x_1\ \forall x\ \{\neg p(x) \vee \{\ \neg\forall y\ [\neg q(x,y) \vee p(f(x_1))] \wedge \forall z\ [\neg q(x,z) \vee p(x)]\ \}\}$$

$$\exists x_1\ \forall x\ \{\neg p(x) \vee \{\exists y\ [q(x,y) \wedge \neg p(f(x_1))] \wedge \forall z\ [\neg q(x,z) \vee p(x)]\ \}\}$$

Conversion to Normal Form

- Step 5: *Push the quantifiers to the right.*

$$\exists x_1 \; \forall x \; \{\neg p(x) \vee \{\exists y \; [q(x,y) \wedge \neg p(f(x_1))] \wedge \forall z \; [\neg q(x,z) \vee p(x)] \}\}$$

$$\exists x_1 \; \forall x \; \{\neg p(x) \vee \{[\exists y \; q(x,y) \wedge \neg p(f(x_1))] \wedge [\forall z \; \neg q(x,z) \vee p(x)] \}\}$$

Conversion to Normal Form

- Step 6: *Eliminate existential quantifiers (Skolemization).*
  - Pick out the leftmost $\exists y\, B(y)$ and replace it by $B(f(x_{i1}, x_{i2}, \ldots, x_{in}))$, where:
    a) $x_{i1}, x_{i2}, \ldots, x_{in}$ are all the distinct free variables of $\exists y\, B(y)$ that are universally quantified to the left of $\exists y\, B(y)$, and
    b) F is any n-ary function constant which does not occur already

Conversion to Normal Form

- Skolemization:

$$\exists x_1 \; \forall x \; \{\neg p(x) \vee \{[\exists y \; q(x,y) \wedge \neg p(f(x_1))] \wedge [\forall z \; \neg q(x,z) \vee p(x)] \}\}$$

$$\forall x \; \{\neg p(x) \vee \{[q(x,g(x)) \wedge \neg p(f(a))] \wedge [\forall z \; \neg q(x,z) \vee p(x)] \}\}$$

Conversion to Normal Form

- Step 7: *Move all universal quantifiers to the left*

$$\forall x \, \{\neg p(x) \vee \{[q(x,g(x)) \wedge \neg p(f(a))] \\ \wedge [\forall z \, \neg q(x,z) \vee p(x)] \}\}$$

$$\forall x \, \forall z \, \{\neg p(x) \vee \{[q(x,g(x)) \wedge \neg p(f(a))] \\ \wedge [\neg q(x,z) \vee p(x)] \}\}$$

Conversion to Normal Form

- Step 8: *Distribute* $\wedge$ over $\vee$.

$$\forall x\ \forall z\ \{[\neg p(x) \vee q(x,g(x))] \wedge [\neg p(x) \vee \neg p(f(a))] \wedge [\neg p(x) \vee \neg q(x,z) \vee p(x)]\ \}$$

- Step 9: (Optional) *Simplify*

$$\forall x\ \{[\neg p(x) \vee q(x,g(x))] \ \wedge\ \neg p(f(a))\ \}$$

Resolution Refutation Proofs

- In refutation proofs, we add the negation of the goal to the set of clauses and then attempt to deduce *False*

Example

- Harry, Ron and Draco are students of the Hogwarts school for wizards
- Every student is either wicked or is a good Quiditch player, or both
- No Quiditch player likes rain and all wicked students like potions
- Draco dislikes whatever Harry likes and likes whatever Harry dislikes
- Draco likes rain and potions
- Is there a student who is good in Quiditch but not in potions?

Resolution Refutation Proofs

Example:

- Jack owns a dog
- Every dog owner is an animal lover
- No animal lover kills an animal
- Either Jack or Curiosity killed the cat, who is named Tuna
- Goal: Did curiosity kill the cat?

- We will add ¬Kills(Curiosity, Tuna) and try to deduce *False*