Complete Knowledge Assumption

Why not assume KB includes all positive facts, and everything else is false (similar to unique name assumption)

Examples:
- If I haven't stated that two rooms are adjacent, assume they are not
- If I haven't stated that Jim is Mary's father, assume he is not

We don't want to state negative facts in KB, but we want to ask queries of form \( \neg p \) and use \( \neg p \) in body of rule

How can we formalize complete knowledge assumption?

Horne Clauses

- Allowed negative information to be expressed
- Some things are true
- Some things are false
- Some things it just doesn't know
- System can conclude now that

Overview

Approach 1

Approach 2

Explicit Quantification

Complete Knowledge Assumption
Example

Variable Case: Example

\( \text{student}(x) \leftarrow x = \text{mary} \)
\( \text{student}(x) \leftarrow x = \text{john} \)
\( \text{student}(x) \leftarrow x = \text{ying} \)

Note that `=` sign could means equality

Collect them all together and you get

\( \text{student}(x) \leftarrow x = \text{mary} \lor x = \text{john} \lor x = \text{ying} \)

Completion is

\( \text{student}(x) \leftrightarrow x = \text{mary} \lor x = \text{john} \lor x = \text{ying} \)

Adding Axioms

If you have

\( a \leftarrow b_1 \ldots a \leftarrow b_n \)

You have equivalently

\( a \leftarrow b_1 \lor \ldots \lor b_n \)

Under the Complete Knowledge Assumption, add in

\( a \rightarrow (b_1 \lor \ldots \lor b_n) \)

If you have predicate \( p \) defined by no clauses in the KB

\( p \rightarrow \text{false} \)

Which is the same as saying

\( \neg p \)

Overview

Complete Knowledge Assumption

Approach 1

Approach 2

Explicit Quantification
Using Clark's Completion

1. Usually also assume UNA
2. Need axioms for equality
3. Need powerful enough proof procedure
4. CLAUS is also assume CNA

Several Clauses

1. Before putting together, make sure each part doesn't have any other variable in common other than the V_i's
2. Do not put C into a number of conjunctions to get C
3. Clark completion is C \rightarrow (V_1=t_1 \land \ldots \land V_n=t_n \land B)
4. Note that only predicates that can be put into Clark's Normal Form can be completed

Variable Case

1. g \land \forall a \forall b \forall c \forall d \forall e \forall f \exists y (g \rightarrow (y=a) \land (y=b) \land (y=c) \land (y=d) \land (y=e) \land (y=f))
Modifying a Top-Down Reasoning Procedure

When see \( \sim p \), do recursive proof:
- If you can prove \( p \) then fail, otherwise succeed.

Example:
- KB = \{ \( p(X) \leftarrow r(X) \land \sim q(X) \), q(a), q(b), r(b), r(d) \}

Negation as Failure

A simpler way to make CKA is to use negation as failure.
- Allow negation as failure in bodies of Datalog clauses.
- Allow negation as failure in queries.
- Call this Negation as Failure.
- A simple way to make the CKA is to use negation as failure.

Overview

- Explicit Quantification
- Complete Knowledge Assumption
- Approach 2
- Approach 1
Quantification

• So far, variables have been universally quantified at clause level
  \( \forall XYZ \) brothers(X,Y) ← (mother(X,Z) \land mother(Y,Z) \land \neg (X = Y))

  - Earlier in course, we argued that it was the same as follows:
    \( \forall XY \) brothers(X,Y) ← (\exists Z mother(X,Z) \land mother(Y,Z) \land \neg (X = Y))

  - Aside: If we assume UNA, equality is the same as unification

  • But how can we capture "every boy loves a girl"

    - This means that for every boy, there exists a girl that the boy likes
      \( \forall X \) boy(X) → (\exists Y girl(Y) \land likes(X,Y))

      \( \forall X (\exists Y girl(Y) \land likes(X,Y)) \leftarrow boy(X) \)

    - Need to explicitly deal with universal and existential quantifiers

Overview

• Complete Knowledge Assumption

• Approach 1

• Approach 2

⇒ Explicit Quantification

VARIABLES

{p, q, r, \neg q, \neg r}

Example:

\[ (X \land \lambda y \phi) \]

When does a global variable exist?
Semantics of Exists

Textbook:

\[ \exists x w \text{ is true in an interpretation } I \iff \text{there is some } d \in \text{domain, say } d \text{ such that if } X \text{ is mapped to } d \text{ than } w \text{ is true in } I \]

Facts about Universal Quantification

Textbook:

\[ \forall x w \text{ is true in an interpretation } I \iff w \text{ is true in } I \text{ regardless of what object in the domain that } X \text{ is mapped to} \]

- Any expression with free variables is same as that expression but with the free variables universally quantified.
- Just making the universal quantification explicit.
- Can prove that order of quantifiers doesn't matter:

\[ \forall x \forall y w \text{ is true exactly when } \forall y \forall x w \text{ is true} \]
Conversion to Clauses

- Step 1: Eliminate → by writing it in terms of ∨, and ¬

- Step 2: Distribute negation so only applies to atomic sentences

- De Morgan's rules for distributing ¬ over ∨ and ∧

- ¬∀vφ replaced by ∃v¬φ and ¬∃vφ replaced by ∀v¬Φ

- Step 3: Rename variables so each quantifier has a unique variable

Existential Elimination

- Example:

  ∃Z block(Z)

  - We know that something is a block.

  - In worst case, there isn't even a name for it in our syntax.

  - So let's make up a new constant, say k, and let k be that block

- Example:

  ∃Z ∃Y hates(Z,Y)

- Example:

  ∀Y ∃Z hates(Y,Z)

  - Who Z is depends on who Y is.

  - Every different value of Y might have a different Z

  - If foe is a new function symbol, we can let foe(Y) be who Y hates

  - So we know that foe(Z) is a function symbol.

  - Example:

    (Z, A) foe(Z, A)

    (Z, B) block(Z)

    - We know something is a block

Examples

- Everyone has a mother

  there is a book that everyone loves

  - There is a book that everyone loves

  - There is a book that everyone loves

  - There is a book that everyone loves

  - There is a book that every boy likes

  - Everyone has a mother
Example Continued

• Step 4: Replace existential quantifiers by skolem functions
  \((\forall x \theta) \land \phi\) becomes \((\forall x \theta) \land \phi\)
  
  Step 5: Universals dropped
  \((\exists x \theta) \lor \phi\) becomes \((\exists x \theta) \lor \phi\)
  
  Step 6: Distribute \lor's over \land's
  \((\exists x \theta) \lor \phi\) becomes \((\exists x \theta) \lor \phi\)
  
  More Steps

\((\forall x \phi) \lor (\exists x \phi)\) becomes \((\forall x \phi) \lor (\exists x \phi)\)

Step 5: Change to disjunction to drop outer universal
  \((\forall x \phi) \lor (\exists x \phi)\) becomes \((\forall x \phi) \lor (\exists x \phi)\)

Step 6: Replace existential quantifiers by skolem functions
Bottom-Up Proof

¬horse(X) ∨ ¬dog(Y) ∨ faster(X,Y) (1)
greyhound(foe) (2)
¬rabbit(Y) ∨ faster(foe,Y) (3)
horse(harry) (4)
rabbit(ralph) (5)
¬greyhound(X) ∨ dog(X) (6)
¬faster(X,Y) ∨ ¬faster(Y,Z) ∨ faster(X,Z) (7)
¬dog(Y) ∨ faster(harry,Y) from 1 & 4 (8)
¬greyhound(Y) ∨ faster(harry,Y) from 8 & 6 (9)
faster(harry,foe) from 9 & 2 (10)
¬faster(foe,Z) ∨ faster(harry,Z) from 10 & 7 (11)
¬rabbit(Z) ∨ faster(harry,Z) from 11 & 3 (12)
faster(harry,ralph) from 12 & 5 (13)
Adding on to Datalog

• Equality
  + Use canonical names and rewrite rules
  + Delay clauses with ungrounded goals

• Inequality
  + Assume UNA and check if objects have different names
  + Delay atoms with inequality if they are not ground

• Arbitrary Clauses
  + Any number of positive or negative literals
  + Include negative answer checks into top-down proof procedures
  + Include answer clauses in KB so can find disjunctive answers

• Negation
  + Make CKA and do embedded proofs
  + Delay atoms with ‘not’ if they are not ground

• Existential Quantification
  + Convert to DCG with skolem functions/variables
  + Convert to CNF with skolem functions/constants