innovate achieve lead

BITS Pilani
Pilani Campus

CS/IS F214
Logic in Computer Science

MODULE: PREDICATE LOGIC

Second-Order Predicate Logic:

Examples

Existential Second-Order Predicate Logic

Second Order Predicate Logic - Examples

- Consider the **Induction Principle** (on *Natural Numbers*):
  - If a property **p** is true for **0**
  - and if **p** is true for n then it is true for **n+1**
  - then **p** is true for all **n**
- We can state this in (First Order) Predicate Logic as

  p(0) ∧ (∀n p(n) --> p(n+1)) --> ∀n p(n)

  but this formula is applicable only for a specific **p**.

- How do we generalize this for any predicate?
  - Quantify over predicates!
    - **∀p (p(0) ∧ (∀n p(n) --> p(n+1)) --> ∀n p(n))**

Second Order Predicate Logic - Examples

- Exercise:
  - State the following in Second Order Logic:
    - Strong Mathematical Induction
    - Structural Induction

Second Order Predicate Logic - Examples

- Suppose you want to state that there exist at two elements (in your universe):
    - How do you state this as a formula ψ_2 in First Order Predicate Logic?
- Is it possible to generalize the formula to state that
    - there exist at least **k** elements, for constant (i.e. fixed) **k**?
-

Second Order Predicate Logic - Examples

- Let ψ_k be the formula stating that there exists at least **k** elements for natural number **k**:
  - What does the set $\{\psi_2, \psi_3, \psi_4, \ldots\}$ indicate?
  - Can you restate the same using a finite set of formulas?
    - In First Order Predicate Logic?
    - In Second Order Predicate Logic?

Second-Order Logic

- Predicate Logic supports ***variables for values*** and ***quantification over variables***
  - but predicate symbols are fixed (i.e. they refer to a single predicate)
    - alternatively, a variable symbol ranges only over values (i.e. *not over predicates*) and
    - consequently predicates are not quantified
- Thus one may refer to **First-Order Predicate Logic**
  - where *predicates are fixed (i.e. they are constants and hence do not require quantification)*
- and **Second-Order Predicate Logic**
  - where *variables may denote predicates (i.e. and hence may be quantified).*

Existential Second-Order Predicate Logic (E-SOPL)

- To address the constraint of *inexpressibility of reachability in graphs* , we attempt an enhancement in our language:
  - support existential quantification of predicates:
    - i.e. we consider formulas ϕ generated by the grammar:
      - $\phi \dashrightarrow \exists \mathbf{p}\, \phi$
      - $\phi \dashrightarrow \psi$
    - where **p** is any predicate symbol and ψ is any (first-order) predicate logic formula.

Models for formulas in E-SOPL

- In First Order Logic, we evaluated a formula φ with respect to a model ***M*** which included
  i. a universe **A** and
  ii. the meaning (or *interpretation*) of each of the function and predicate symbols used in φ
- In E-SOPL, a model has to include
  - a set of possible meanings (or *interpretations*) for each variable predicate .

Models for formulas in E-SOPL

- For instance, consider the following formula
  - **$\exists q\ (q(0) \wedge (\forall n\ q(n) \dashrightarrow q(succ(n))) \dashrightarrow \forall n\ q(n))$**
- A suitable model M in E-SOPL would include (in addition to (i) and (ii)):
  - $q^M \subseteq$ **P**(A) (i.e. ps^M is a set of subsets of A – Why?)
    - q^M must be non-empty. (Why?)

  [Note: q is a variable – it can be any unary relation on A i.e. any meaning of q is a subset of A. End of Note.]

E-SOPL: Semantics

- Given a suitable model ***M***, and a look-up table ι , a formula $\exists$**p** φ can be evaluated as follows:
  - $\boldsymbol{M} \mid=_{\iota} \exists \mathbf{p}\, \varphi$ iff for some for R in ps^M , $\boldsymbol{M} \mid=_{\iota\, [p|\text{->}R]} \varphi$

Expressing Properties of the Path Relation

- Express the properties of a Path relation:
  - Path relation is reflexive
    - **REF $\equiv_{def}$ $\forall$X p(X,X)**
  - Path relation is transitive
    - **TRANS $\equiv_{def}$ $\forall$X $\forall$Y $\forall$Z p(X,Y) $\wedge$ p(Y,Z) --> p(X,Z)**
  - Path relation is implied by the edge relation **E** in a graph.
    - **EDGE_PATH $\equiv_{def}$ $\forall$X $\forall$Y E(X,Y) --> p(X,Y)**

- Can we combine these formulas to express <u>reachability</u> in directed graphs?

Expressing Un-reachability

- Combine the formulas for properties of the PATH relation to express un-reachability in directed graphs:
  - $\exists p$ REF $\wedge$ TRANS $\wedge$ EDGE_PATH $\wedge \neg p(u,v)$

  or

  - $\exists p\ (\forall X\ p(X,X)) \wedge (\forall X\ \forall Y\ \forall Z\ p(X,Y)) \wedge p(Y,Z) \dashrightarrow p(X,Z)) \wedge (\forall X\ \forall Y\ E(X,Y) \dashrightarrow p(X,Y)) \wedge \neg p(u,v)$

- This states that:
  - there is a relation p that is reflexive and transitive, and is implied by the edge relation E in a graph
  - such that *vertices u and v are not related by p*

Expressing Reachability

- The formula for un-reachability can be negated to express reachability:
  - i.e. v is reachable from u can be specified as:
    - $\neg (\exists p\ REF \wedge TRANS \wedge EDGE\_PATH \wedge \neg\, p(u,v))$
- Is this formula in existential second-order logic?
  - Why or why not?

Expressing Reachability in Universal Second-Order Logic

- The formula for un-reachability can be negated to express reachability:
  - Is this formula in existential second-order predicate logic?
    - Recall that, in first-order predicate logic:

      $\neg (\exists X\ \phi) \equiv \forall X \neg \phi$
    - Analogously,

      $\neg(\exists p\ REF \wedge TRANS \wedge EDGE\_PATH \wedge \neg\ p(u,v))$

      $\equiv\ \forall p \neg (REF \wedge TRANS \wedge EDGE\_PATH) \vee p(u,v)$

      $\equiv \forall p\ \ (REF \wedge TRANS \wedge EDGE\_PATH) \dashrightarrow p(u,v)$

  - This formula is in **universal second-order predicate logic**!

Universal Second-Order Predicate Logic

- Define Universal Second-Order Predicate Logic as the language whose formulas ϕ are generated by the grammar:
    - ϕ --> $\forall$**p** ϕ
    - ϕ --> ψ
  - where p is any predicate symbol and ψ is any (first-order) predicate logic formula.

- Given a suitable model ***M***, and a look-up table ι , a formula $\forall$**p** φ can be evaluated as follows:
  - ***M*** $|=_\iota \forall \mathbf{p}\, \varphi$ iff for all R in ps^M, ***M*** $|=_{\iota\ [p|\text{->}R]} \varphi$