Artificial Intelligence
Chapter 15
The Predicate Calculus

Biointelligence Lab
School of Computer Sci. & Eng.
Seoul National University
Outline

- Motivation
- The Language and Its Syntax
- Semantics
- Quantification
- Semantics of Quantifiers
- Predicate Calculus as a Language for Representing Knowledge
- Additional Readings and Discussion
15.1 Motivation

- **Propositional calculus**
  - Expressional limitation
  - Atoms have no internal structures.

- **First-order predicate calculus**
  - has names for objects as well as propositions.
  - Symbols
    - Object constants
    - Relation constants
    - Function constants
  - Other constructs
  - Refer to objects in the world
  - Refer to propositions about the world
15.2 The Language and its Syntax

- Components
  - Infinite set of *object constants*
    - Aa, 125, 23B, Q, John, EiffelTower
  - Infinite set of *function constants*
    - fatherOf, distanceBetween, times
  - Infinite set of *relation constants*
    - B17, Parent, Large, Clear, X11
  - Propositional connectives
    - \( \lor, \land, \neg, \subseteq \)
  - Delimiters
    - (, ), [, ], (separator)
15.2 The Language and its Syntax

- **Terms**
  - Object constant is a term
  - Functional expression
    - fatherOf(John, Bill), times(4, plus(3, 6)), Sam

- **wffs**
  - Atoms
    - Relation constant of arity $n$ followed by $n$ terms is an atom (atomic formula)
    - An atom is a wff.
    - Greaterthan(7,2), P(A, B, C, D), Q
  - Propositional wff
    - $[\text{Greaterthan}(7,2) \land \text{Lessthan}(15,4)] \lor \neg \text{Brother}(\text{John}, \text{Sam}) \lor P$
15.3 Semantics

- Worlds
  - Individuals
    - Objects
    - Concrete examples: Block A, Mt. Whitney, Julius Caesar, …
    - Abstract entities: 7, set of all integers, …
    - Fictional/invented entities: beauty, Santa Claus, a unicorn, honesty, …
  - Functions on individuals
    - Map $n$ tuples of individuals into individuals
  - Relations over individuals
    - Property: relation of arity 1 (heavy, big, blue, …)
    - Specification of $n$-ary relation: list all the $n$ tuples of individuals
15.3 Semantics (Cont’d)

- Interpretations
  - Assignment: maps the followings
    - object constants into objects in the world
    - \( n \)-ary constants into \( n \)-ary functions
    - \( n \)-ary relation constants into \( n \)-ary relations
    - called *denotations* of corresponding predicate-calculus expressions
  - Domain
    - Set of objects to which object constant assignments are made
  - *True/False* values

Figure 15.1 A Configuration of Blocks
<table>
<thead>
<tr>
<th>Predicate Calculus</th>
<th>World</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>A</td>
</tr>
<tr>
<td>B</td>
<td>B</td>
</tr>
<tr>
<td>C</td>
<td>C</td>
</tr>
<tr>
<td>F1</td>
<td>Floor</td>
</tr>
<tr>
<td>On</td>
<td>On={&lt;B,A&gt;, &lt;A,C&gt;, &lt;C, Floor&gt;}</td>
</tr>
<tr>
<td>Clear</td>
<td>Clear={&lt;B&gt;}</td>
</tr>
</tbody>
</table>

Table 15.1 A Mapping between Predicate Calculus and the World

Determination of the value of some predicate-claculus wffs

- `On(A,B)` is *False* because `<A,B>` is not in the relation `On`.  
- `Clear(B)` is *True* because `<B>` is in the relation `Clear`.  
- `On(C,F1)` is *True* because `<C,Floor>` is in the relation `On`.  
- `On(C,F1) ∧ ¬On(A,B)` is *True* because both `On(C,F1)` and `¬ On(A,B)` are True.
15.3 Semantics (Cont’d)

- Models and Related Notions
  - An interpretation \textit{satisfies} a wff
    - wff has the value \textit{True} under that interpretation
  - \textbf{Model} of wff
    - An interpretation that satisfies a wff
  - \textbf{Valid} wff
    - Any wff that has the value \textit{True} under \textit{all} interpretations
  - \textit{inconsistent/unsatisfiable} wff
    - Any wff that does not have a model
  - \textit{\Delta logically entails \omega} (\Delta \models \omega)
    - A wff \omega has value \textit{True} under all of those interpretations for which each of the wffs in a set \Delta has value \textit{True}
  - \textbf{Equivalent wffs}
    - Truth values are identical under \textit{all} interpretations
15.3 Semantics (Cont’d)

- Knowledge
  - Predicate-calculus formulas
    - represent knowledge of an agent
  - Knowledge base of agent
    - Set of formulas
    - The agent knows $\omega = \text{the agent believes } \omega$

Figure 15.2 Three Blocks-World Situations
15.4 Quantification

- **Finite domain**
  - \( \text{Clear(B1) \land Clear(B2) \land Clear(B3) \land Clear(B4)} \)
  - \( \text{Clear(B1) \lor Clear(B2) \lor Clear(B3) \lor Clear(B4)} \)

- **Infinite domain**
  - Problems of long conjunctions or disjunctions \( \rightarrow \) impractical

- **New syntactic entities**
  - **Variable symbols**
    - consist of strings beginning with lowercase letters
    - term
  - **Quantifier symbols** \( \rightarrow \) give expressive power to predicate-calculus
    - \( \forall \): universal quantifier
    - \( \exists \): existential quantifier
15.4 Quantification (Cont’d)

- $(\forall \zeta)\omega, (\exists \zeta)\omega$ : wff
  - $\omega$: wff $\Rightarrow$ within the scope of the quantifier
  - $\zeta$: quantified variable

- Closed wff (closed sentence)
  - All variable symbols besides $\zeta$ in $\omega$ are quantified over in $\omega$
    
    $(Ax)[P(x) \supset R(x)], \ (\exists x)[P(x) \supset (Ey)[R(x, y) \supset S(f(x))]]$
  - Property
    
    
    $(\forall x)[(\forall y)\omega] = (\forall y)[(\forall x)\omega] = (\forall x, y)\omega$
    
    $(\forall x)[(\exists y)\omega] \neq (\exists y)[(\forall x)\omega]$

- First-order predicate calculi
  - restrict quantification over relation and function symbols
15.5 Semantics of Quantifiers

- Universal Quantifiers
  - $(\forall \zeta) \omega(\zeta) = True$
    - $\omega(\zeta)$ is True for all assignments of $\zeta$ to objects in the domain
  - Example: $(\forall x)[On(x,C) \supset \neg Clear(C)]$ in Figure 15.2
    - $x$: A, B, C, Floor
    - investigate each of assignments in turn for each of the interpretations

- Existential Quantifiers
  - $(\exists \zeta) \omega(\zeta) = True$
    - $\omega(\zeta)$ is True for at least one assignments of $\zeta$ to objects in the domain
15.5 Semantics of Quantifiers (Cont’d)

- Useful Equivalences
  - $\neg(\forall \xi)\omega(\xi) \equiv (\exists \zeta)\neg\omega(\zeta)$
  - $\neg(\exists \xi)\omega(\zeta) \equiv (\forall \zeta)\neg\omega(\zeta)$
  - $(\forall \xi)\omega(\zeta) \equiv (\forall \eta)\omega(\eta)$

- Rules of Inference
  - Propositional-calculus rules of inference $\rightarrow$ predicate calculus
    - *modus ponens*
    - Introduction and elimination of $\land$
    - Introduction of $\lor$
    - $\neg$ elimination
    - Resolution
  - Two important rules
    - Universal instantiation (UI)
    - Existential generalization (EG)
15.5 Semantics of Quantifiers (Cont’d)

- **Universal instantiation**
  - $(\forall \xi)\omega(\xi) \rightarrow \omega(\alpha)$
  - $\omega(\xi)$: wff with variable $\xi$
  - $\alpha$: constant symbol
  - $\omega(\alpha)$: $\omega(\xi)$ with substituted for $\xi$ throughout $\omega$
  - Example: $(\forall x)P(x, f(x), B) \rightarrow P(A, f(A), B)$

- **Existential generalization**
  - $\omega(\alpha) \rightarrow (\exists \xi)\omega(\xi)$
  - $\omega(\alpha)$: wff containing a constant symbol $\alpha$
  - $\omega(\xi)$: form with $\xi$ replacing every occurrence of $\alpha$ throughout $\omega$
  - Example: $(\forall x)Q(A, g(A), x) \rightarrow (\exists y)(\forall x)Q(y, g(y), x)$
15.6 Predicate Calculus as a Language for Representing Knowledge

- Conceptualizations
  - Predicate calculus
    - language to express and reason the knowledge about real world
    - represented knowledge: explored throughout logical deduction
  - Steps of representing knowledge about a world
    - To conceptualize a world in terms of its objects, functions, and relations
    - To invent predicate-calculus expressions with objects, functions, and relations
    - To write wffs satisfied by the world: wffs will be satisfied by other interpretations as well
15.6 Predicate Calculus as a Language for Representing Knowledge (Cont’d)

- Usage of the predicate calculus to represent knowledge about the world in AI
  - John McCarthy (1958): first use
    - CYC project
    - represent millions of commonsense facts about the world
  - Nilsson 1991: discussion of the role of logic in AI
  - Genesereth & Nilsson 1987: a textbook treatment of AI based on logic
15.6 Predicate Calculus as a Language for Representing Knowledge (Cont’d)

- **Examples**
  - Examples of the process of conceptualizing knowledge about a world
  - Agent: deliver packages in an office building
    - **Package**\( (x) \): the property of something being a package
    - **Inroom**\( (x, y) \): certain object is in a certain room
    - Relation constant **Smaller**\( (x, y) \): certain object is smaller than another certain object
    - “All of the packages in room 27 are smaller than any of the packages in room 28”

\[
(\forall x, y) \{ [\text{Package}(x) \land \text{Package}(y) \land \text{Inroom}(x, 27) \land \text{Inroom}(y, 28)] \Rightarrow \text{Smaller}(x, y) \}
\]
"Every package in room 27 is smaller than one of the packages in room 29"

\[(\exists y)(\forall x)\{[\text{Package}(x) \land \text{Package}(y) \land \text{Inroom}(x, 27) \land \text{Inroom}(y, 28)] \supset \text{Smaller}(x, y)\}\]

\[(\forall x)(\exists y)\{[\text{Package}(x) \land \text{Package}(y) \land \text{Inroom}(x, 27) \land \text{Inroom}(y, 28)] \supset \text{Smaller}(x, y)\}\]

Way of stating the arrival time of an object

- \text{Arrived}(x, z)
- \text{X}: arriving object
- \text{Z}: time interval during which it arrived
- "Package A arrived before Package B"
  \[(\exists z_1, z_2)[\text{Arrived}(A, z_1) \land \text{Arrived}(B, z_2) \land \text{Before}(z_1, z_2)]\]
- Temporal logic: method of dealing with time in computer science and AI
Difficult problems in conceptualization

- “The package in room 28 contains one quart of milk”
  - Mass nouns
  - Is milk an object having the property of being whit?
  - What happens when we divide quart into two pints?
  - Does it become two objects, or does it remain as one?

Extensions to the predicate calculus

- allow one agent to make statements about the knowledge of another agent
  - “Robot A knows that Package B is in room 28”
Additional Readings

- McDermott & Doyle 1980: discussion about
  - the use of logical sentences to represent knowledge
  - the use of logical inference procedures to do reasoning

- Tarski 1935, Tarski 1956: Tarskian semantics
  - Controversy about mismatch between the precise semantics of logical languages

- Agre & Chapman 1990
  - Indexical functional representations

- Enderton 1972, Pospesel 1976
  - Boos on logic

- Barwise & Etchemendy 1993
  - Readable overview on logic