# Knowledge Representation and Reasoning

Predicate Calculus
and Resolution

# Predicate Calculus

- What is Predicate Calculus (Pd.C.)?
- Elements of Pd.C. – Formal Language
- Semantics and Quantifiers
- Unification and Resolution
- Resolution as Inference Rule
- Pd.C. for Knowledge Representation

# Motivation for Pd.C.

- Propositional Calculus (P.C.) is not flexible
- Each atom is completely different from another
- There are no “groups” of atoms with common properties or references to the real world
- A more generic approach is needed for encoding objects and propositions

# Formal Pd.C. Language

## First-order Predicate Calculus (language):

- Components:
  - Object constants: “Aa”, “123”, “John”, ...
  - Function constants: “Parent”, “FatherOf”, “DistanceBetween”, ...
  - Relation constants: “Parent”, “Large”, “Clear”, ...
  - Connectives ( $\wedge$ , $\vee$ , $\supset$ , $\neg$ ) and Delimiters ( ( , ) , [ , ] )
- Terms:
  - Object constants: “Aa”, “Charlie”, ...
  - Function constants: “FatherOf(John,Bill)”, ...
- Well-formed formulas (**wff**):
  - Atoms: any Relation constant containing any number of Terms
  - Propositional wff: logical combinations of Atoms

# Pd.C. Semantics

- Worlds:
  - Can have an infinite number of Individuals (Objects)
  - Infinite number of Functions can be applied to Individuals
  - Arbitrary number of Relations over Individuals
- Interpretations:
  - Pd.C. expression that maps Object constants to the World
  - Includes Individuals, Functions, Relations
- Models:
  - Is an Interpretation that satisfies a given wff
  - Usually one or more instances of a Function application
- Knowledge:
  - A set of wff that describe the World at a given state

# Pd.C. Quantification

- Variable symbols (Variables):
  - Denoted by lowercase letters (example: “x”, “p1”, ...)
  - Used instead of large groups of Objects sharing some common attribute
  - Can be used with a Function or Relation constant

- Quantifier symbols:
  - Universal Quantifier: $\forall$ (“for-each”)
  - Existential Quantifier: $\exists$ (“exists”)
  - If Q is either $\forall$ or $\exists$ , “(Qx)$\omega$” means x is quantified within wff $\omega$

# Semantics of Quantifiers

Rules of Inference:

- Universal Instantiation (UI):
  - $(\forall x)wff(x) \rightarrow wff(A)$
  - Example: $(\forall x)Parent(x,from(x),B) \rightarrow Parent(A,from(A),B)$
- Existential Generalization (EG):
  - $wff(A) \rightarrow (\exists x)wff(x)$
  - Example: $Child(B,by(A),A) \rightarrow (\exists y)Child(y,by(A),A)$

Note: both UI and EG are “sound” inferential rules

# Simplification of wff

- **De Morgan's Laws**:

  $\neg (A \vee B) \equiv \neg A \wedge \neg B$

  $\neg (A \wedge B) \equiv \neg A \vee \neg B$

- Contraposition:

  $\neg (A \supset B) \equiv (\neg A \supset \neg B)$

- Equivalences in Quantifiers:

  $\neg (\forall A) wff(A) \equiv (\exists A) \neg wff(A)$

  $\neg (\exists A) wff(A) \equiv (\forall A) \neg wff(A)$

  $(\forall x) wff(x) \equiv (\forall y) wff(y)$

# Unification and Resolution

Unification:

- A literal may contain variables and quantifiers
- The World contains a limited set of appropriate Identities
- Unification replaces universal quantifiers ($\forall$) with real Atoms
- New set of instances (Models) can be resolved

Resolution:

- Apply Unification in the most general way, i.e. to include as many valid Atoms of the World as possible
- Convert to Clausal form and apply Resolution as in Propositional wff.

# Resolution in Pd.C.

- Resolution: use **deduction** rules to assert or discard the validity of a Clause.
- Clause: any formatted wff that is used in a Resolution scheme.
- Resolution on Clauses:
  - Follow 6 simple rules for converting any wff into a clause in CNF that can be resolved

# Resolution in Pd.C.

- Converting a clause and resolve:

  $\neg(P \supset Q) \vee (R \supset P)$

1. Eliminate implication signs (⊃) by using the equivalent form using (¬∨):

   $\neg(\neg P \vee Q) \vee (\neg R \vee P)$

2. Reduce the (¬) signs using De Morgan's laws:

   $(P \wedge \neg Q) \vee (\neg R \vee P)$

3. Eliminate existential quantifiers (∃):

   $\neg (\exists P) wff(P) \equiv (\forall P) \neg wff(P)$

Note: step 3 includes a substitution stage via Skolem functions

# Resolution in Pd.C.

4. Convert to CNF, i.e. place ($\wedge$) outside parentheses:

   $\neg (A \wedge B) \equiv \neg A \vee \neg B$

5. Eliminate universal quantifiers ($\forall$):

   $(\forall x)P(x) \equiv \{P(A),P(B),...\}$

6. Convert to wff, i.e. remove ($\wedge$) connectives:

   $(P\vee\neg R\vee P) \wedge (\neg Q\vee\neg R\vee P) \equiv \{(P\vee\neg R) , (\neg Q\vee\neg R\vee P)\}$

Note: Final resolution result contains all instances of the initial Clause that evaluate to True

# Resolution as an Inference Rule

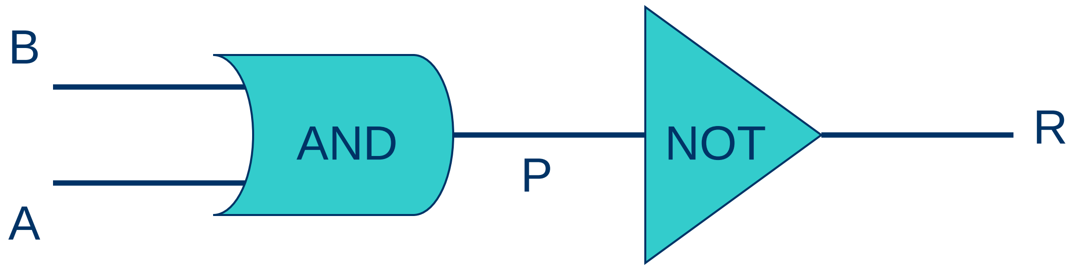
- **Resolution Refutation**: proof of the negation (non-empty result) invalidates the original clause, otherwise the original clause is asserted as true.

- Proving: formulate the problem as a set of Facts that describe the World, then describe the (unknown) solution as a Clause to be Resolved (True or False).
- Answering: do the same as in Proving, only here we also want to keep track of the literals that satisfy the proven assertion by the Resolution process ("answers").

Note: Resolution Refutation as Inference Rule is both "sound" and "complete" (Robinson, 1965).

# An Exercise

Use a compact set of Atoms and Clauses to describe a World for the following Boolean circuit:




Hint: Describe the Boolean gates as Functions ("AND", "NOT") and the wire signals as Atoms of the same base Object ("Signal").

# P.C. – Readings


- Nils J. Nilsson, “Artificial Intelligence – A New Synthesis”, Morgan Kaufmann Publishers (1998). [see: ch.15 & ch.16]

- S. J. Russell, P. Norvig, “Artificial Intelligence: A Modern Approach”, 2nd/Ed, Prentice Hall, 2002.