Day 10B: Logical Representation

- For a long time, first order predicate calculus was the gold standard for representing knowledge in AI.
- What is predicate calculus?
  - A language for expressing facts and rules, and a mathematics for manipulating truth values and well-formed expressions
- What is logic used for in AI?
  - Representation of problem states and rules
  - Reasoning
  - Data retreival
  - Expert systems
  - Inferences
- Three brands of inference:
  - Abduction
    - Approximate reasoning and plausible inference
    - If P implies Q, and Q is true, abduce (infer) P is true
    - This is not logically sound, but a good heuristic
  - Induction
    - To reason from specific cases to general rules
    - Very difficult to mechanize, a major problem to be solved with machine learning
    - Ex: If you know that robins fly, sparrows fly, and hawks fly, then infer that all birds fly
  - Deduction (à la Sherlock)
    - Given a set of:

        - Assumptions/Facts
        - Logical Rules
        - Universal Laws of Legic
    - Find all new facts which logically follow from our assumptions
- Logical expressions are built out of primitives, according to rules of well-formedness:
    - Constants: symbols which represent specific objects in the real world (ex. `MARY`, `JOHN`)
    - Variables: indefinite/nonspecific references to objects and individuals in the real world (ex. `x`, `y`)
    - Predicates: functions that denote aspects of objects or relationships between objects (ex. `LOVES(JOHN, MARY)`, `RED(BALL)`, etc.)
    - Connectives: logical operators which compute or constrain truth values:
        - AND: &
        - OR: |
        - NOT: ~
        - IMPLIES: ⇒
            - Truth Table: P ⇒ Q:
                - If P is true, Q is true
                - If P is false, nothing is implied
                - If Q is true, nothing is implied (?)
                - If Q is false, P is false
    - Quantifiers: Logical meta-operators which assert across the entire population of objects
        - For all: ∀
        - There exists: ∃
- Other logical constructions:

    - Literals: well-formed expressions which state facts about specific individuals (no quantifiers or variables):
        - `LOVES(MARY, JOHN) & MARRIED(JOHN)`
        - `MALE(TERRY) & ~MARRIED(TERRY)`
    - Formuale: well-formed expressions which assert generalized rules about relationships between objects
- Propositional Logic vs. Predicate Logic
    - Propositional Logic works with constants, while predicate logic works with functions
- How to use formal logic?
    - Decide on objects and predicates
    - Translate facts and rules about the domain to expressions
    - Then mechanically solve it
- Old AI extremist position?
    - All human thought can be reduced to logic
    - All logic can be mechanized
    - Therefore all human thought can be mechanized through logic
    - This is completely wrong, because solving logical equations is NP-complete, and humans are somehow solving it faster.
        - (Ari's addition) Also, people are sometimes illogical
- Logic is already implicit in programs, but in AI it is made more explicit
- Problems with using logic remain:
    - Setting up rules and predicates is hard and subjective
    - Ambiguity!
    - Mechanics:
        - Combinatorial Explosion
        - Some rules are infinitely recursive

- Some mechanics of logic:
  - Double negation: `~~P = P`
  - DeMorgan's Laws: `~(P & Q) = (~P | ~Q)`, `~(P | Q) = (~P & ~Q)`
    - Also for quantification: $!\exists x P(x) = \forall x !P(X)$, $!\forall x Q(x) = \exists x !Q(x)$
  - Commutativity: `P & Q = Q & P`, `P | Q = Q | P`
  - Distributivity: `X & (Y | Z) = (X & Y) | (X & Z)`, `X | (Y & Z) = (X | Y) & (X | Z)`
  - Associativity: `(A & B) & C = A & (B & C)`
  - Modus Ponens (Forward): If `P => Q` and `P = true`, then `Q = true`
  - Modus Tollens (Backward): If `P => Q` and `Q = false`, then `P = false`
  - Syllogism: If `P => Q & Q => R` then `P => R`
  - Use existing problem-solving techniques (BFS/DFS, etc.) to search through the problem space of formal logic
- Combinatorial Explosion:
  - Some rules are infinitely recursive (ex. Every person has a parent, including that parent), and it's unclear how much depth you need
  - Lots of operators and lots of rules means there are lots of true facts, which results in too many options
  - Solved by Resolution
- Resolution: The rule to end all rules
  - One way to represent things: Conjunctive Normal Form: a series of ORs AND'd together
  - One inference rule: Disjunctive Syllogism (KNOW THIS NAME FOR THE MIDTERM)
    - Two Disjuncts yield something new, ex: `(A | B | C) & (D | ~B | E) = (A | C | D | E)` (Bs cancel out)
    - Subsumes other laws of inference:
      - Modus Ponens: If `P` and `(~P | Q)` then `Q`

- Modus Tollens: If `~Q`, and `(~P | Q)` then `~P`
- Chaining