Overview

- Formal logic deeply influences alternative approaches to theorizing about the human cognition
    - Despite not being used itself as a framework

Rules
- One such framework that borrows from formal logic

<u>Definition:</u> A rule is an IF-THEN structure that is modelled on logical implication, but defined somewhat differently
- Seems suited to represent knowledge
- They are an attempt to adapt formal logic in a way to retain the powers of formal logic
    - Also addresses its weaknesses as a framework for theories of cognition
- Main strength of using formal logic is the way its symbolization system permits the representation of the content of sentences and links between sentences
    - Ex:

$(\forall x)(Gx \supset Bx)$	All geese are birds.
$\underline{(\forall x)(Bx \supset Wx).}$	<u>All birds have wings.</u>
$(\forall x)(Gx \supset Wx)$	All geese have wings.

- Problem with using formal logic is that it is specialized solely to representing arguments
    - Language is more general purpose, and by extension so should our cognition
- A rule-based framework borrows heavily from the representational power of formal logic, but modify it in certain ways
    - Meaning of IF-THEN is broadened
    - Application of strategies is limited by search strategies

Rules-Based Systems
- Typically consists of three components
    - Representation of a goal and initial condition
    - Database of rules (Can be thought of as knowledge)
    - Strategy or algorithm to apply the rules in order to achieve the goal

<u>Goal:</u> A state of affairs the problem solver seeks to reach

<u>Initial condition:</u> State of affairs the problem solver starts off at
- Both can be represented as a list of propositions that are true in that particular state

Ex: Tower of Hanoi

- In artificial intelligence, a rule is called a <u>production.</u>
    - Thus a rule-based system can also be called a production system

Search
- <u>Definition:</u> The way to determine which rules to combine in what order
- A search through a database of rules is similar to a physical search
    - We know where the goal is, but not exactly where

- Process of searching for a goal is to find a route from where you are to where the goal is
- Could be random, or systematic
- Method for organizing a search is known as a search strategy
    - Constructing a solution to a problem by combining database rules can also be thought of as a search
        - Called a knowledge search by Newell
- The search area of a knowledge search is known as the knowledge space
    - Set a situations connected together through rules of database
    - Each location is a situation that is able to be brought up from the rules
    - Neighbouring locations are states that can be reached by applying a rule

Search Strategies

- Forward Strategy
    - Current state is compared to antecedent conditions (IF part) to determine if there's a match
    - Resembles rule of **modus ponens** of formal logic
        - From a statement p, we can apply the rule p->q to conclude:
            - $$\begin{array}{l} p \supset q \\ \underline{p \quad .} \\ q \end{array}$$
- Backward Strategy
    - Goal is compared to the actions (THEN part) to determine if any rules in the database satisfy the goal
        - If it does, continue the search by using the antecedent part of the rule as a new goal - a subgoal
            - Then loop
        - Once we reach a point where antecedent == initial conditions, we're done
    - Analogous to the fallacy of affirming the consequent in formal logic
        - That is, given a statement q and p->q, we can fallaciously conclude:
            - $$\begin{array}{l} p \supset q \\ \underline{q \quad .} \\ p \end{array}$$
        - May appear worrying that it is non-valid in formal logic, but forward and backward strategies should not be treated as valid and non-valid arguments in formal logic
- Bidirectional strategy
    - Use both strategies depending on which is more appropriate

Heuristics

- The above strategies could be used in a large number of ways
    - Known as heuristics
- Guides the execution of a search strategy in order to (hopefully) find the most efficient path to a goal

- Depth-first search:
    - Apply the first rule whose antecedent matches initial condition
        - Loop
    - Requires traveling through a given path until a solution is found, or backtracking if none is found
- Breadth-first search:
    - Every rule whose antecedent matches the initial condition is applied, and all are matched in the next loop
- Best-first search:n
    - Every rule whose antecedent matches initial condition is ranked with a function
        - Evaluates how likely that a rule would lead to a solution
            - LT program did this, would determine which rule would reduce the difference between goal and initial state the most

Which strategy to use

- Depends on how the solutions are distributed in the search space
- Ex: Tic Tac Toe
    - Many goal states relative to the total number of states
        - Easy to reach a goal state if we follow an arbitrary initial
            - If not, we can easily find another goal state
    - Best way is to consider the rules in a fixed order
        - Rank the rules in order to determine which one you want to consider first
            - This is a best-first approach
            - For TTT, we don't need to consider context, these rules are simple
- Harder for games like chess
    - Large selection of opening moves - computers and humans have learned a sequence of them that compress much of the depth into a smaller set
    - Therefore, we forward chain with breadth-first heuristic
    - Midgame has a much larger midgame, hard to compress, goals are too far ahead
        - Players adopt subgoals to achieve strong positions
            - Therefore, we backward chain from these subgoals
    - Endgame
        - Smaller search space, more goal states
            - Backward chain with depth-first search
- Strategies change with expertise
    - Ranking of rules may change based on context
        - An expert may use best-first more often than novices
    - Suggests that framing theories about intelligence with a rule-based system could be logical
        - Rules seem to be a natural way to capture knowledge
            - Knowledge seems to be central to intelligence

Evaluation of Rules

- What are the strengths and weaknesses of a rule-based system as a framework?
    - Are they a good representation of knowledge?
    - Is "Search" a good procedure to model intelligent thinking?

Representational Power

- Recall that a problem with formal logic is that it is highly specialized in representing statements and validity of arguments
    - Natural languages are more flexible, implying that people's thoughts would likewise be flexible
    - Rule-based systems, although similar to formal logic is defined differently to allow greater flexibility
- Ex:
  1. $(\forall x)(Bx \supset Fx)$ "All birds fly."
  2. IF x is a bird THEN x flies.
    - This logical implication says that all birds without exception must fly
    - A rule however, would say something like "Usually, if something is a bird, it would be able to fly."
    - The implication would be false if there were an exception, like a penguin, but would still hold true for a rule
    - Exceptions are a disaster in formal logic, would result in a contradiction from valid arguments
        - A rule-based system can simply decide to retract one of the conclusions that would lead to a contradiction, perhaps using a heuristic to determine which is more preferable
        - A rule-based system doesn't commit to any rules in its database
- Rules are viewed as **defaults -** true unless an exception says otherwise
- Valid arguments in formal logic are <u>monotonic - </u>number of conclusions doesn't decrease
    - In rule-based system, it is non-monotonic - conclusions can be retracted
    - Therefore, in a rule-based system, we are not engaging in deduction
        - We are not solely proceeding with valid arguments
        - The basic form of inference is reasoning instead of deduction
            - Deduction therefore has little to do with rule-based systems

Computational Power

- **Explanation**
    - Forward and backward chaining in rule-based systems can be used to model explanations
- Forward chaining works similarly to deductive explanation
    - IF _____ then ______
        - The reverse would be like abductive explanation
            - Given ____, assume antecedent

    - However, if there are two IFs with different conclusions, then it may be difficult to find the right explanation
        - We can rank each rule, but it's difficult to know this in advance
            - Common in expert systems
    - Further limitations w/ expert system explanations:
        - The explanation usually involves how the conclusion was reached - a <u>trace</u>
            - However, a trace could be a bad explanation
                - Ex: Given these symptoms, a patient has this disease, but why?
- **Learning**
    - A system may be able to learn new rules, similar to how we learn as we progress in life
    - Learning of new rules can be facilitated with:
    - <u>Inductive Generalization</u>
        - Occurs when two things are frequently conjoined, so that one implies the other
    - <u>Chunking (SOAR) or Composition (ACT-R)</u>
        - Chunking refers to putting together a string of separate representations that occur frequently together
            - Rules applied often in succession can be chunked into one rule
    - <u>Specialization</u>
        - If a certain rule doesn't yield good results in certain circumstances, make a new rule to deal with the exception
- **Language**
    - Language is generative, we have a set of rules that assign structure to a sentence, in order to determine if a sentence if valid or not
    - Syntax of words can be modelled by rules
        - Chomsky believed the same to be true of morphology and phonology of words
    - Pinker points out that some elements of morphology is better explained with an <u>associationist model</u>
        - Ex: Sing is not singed, but sang
            - We learn this from similar examples, like "Ring"
            - System notes that the problem is similar to another one with a solution, and assumes that their solutions are similar

Limitations

- Although a rule-based system may simulate human expertise in one area well, how can it simulate all facets of human intelligence?
    - Scale up the system
        - Add all the rules in a given expertise together into a database
            - If expertise = rules + search strats

    - This approach is currently impossible due to the <u>scaling problem</u>
        - When a rule database gets larger, the conflicts b/w rules also increase
            - Leading to incompatible conclusions or exceptions
    - The effort required to manage all these conflicts takes away from actually applying the rules
        - Called <u>thrashing</u>

Psychological Plausibility

- There exist differences between the performance of experts vs novices that the rule-based models emulate well
    - Experts usually apply <u>domain-specific</u> knowledge
        - Performance is usually automatic, quick
        - Acquired through training/repetition
    - Novices use domain-independent knowledge, more broad methods
        - Performance is slower, more tentative
        - More mental effort required
- In humans, an increase in performance with training displays the <u>power law of practice</u>
    - Each training session produces less gain than the previous (Diminishing returns)
    - Can be explained with chunking
        - Adding knowledge through chunking a rule system is easy in the beginning but gets harder as exceptions to the rule become rarer

Philosophical Issues

- Chunking together knowledge, according to Allen Newell is essential to human intelligence
    - No intelligent being can have a plan ready for every problem
        - We must combine knowledge
    - However, we must constrain the search by finding the best strategy in a given scenario
    - This is the classic view on Cognitive Science, where intelligence is having knowledge and having the means of using it
- **Problems**
    - Frame problem
        - It is hard for a rule-based system to distinguish between relevant rules and non relevant rules to a given problem