Problem Characteristics

1. Is the problem decomposable?
2. Can solution steps be ignored or undone?
3. Is the universe predictable?
4. Is a good solution absolute or relative?
5. Is the solution a state or a path?
6. What is the role of knowledge?
7. Does the task require interaction with a person?

Production system

It is useful to structure AI programs in a way that facilitates describing and performing the search process. Production system provides such structures In other words the process of solving the problem can usefully be modeled as production system

A production system consists of four basic components:

a. A set of rules of the form Ci → Ai where Ci is the condition part and Ai is the action part. The condition determines when a given rule is applied, and the action determines what happens when it is applied.

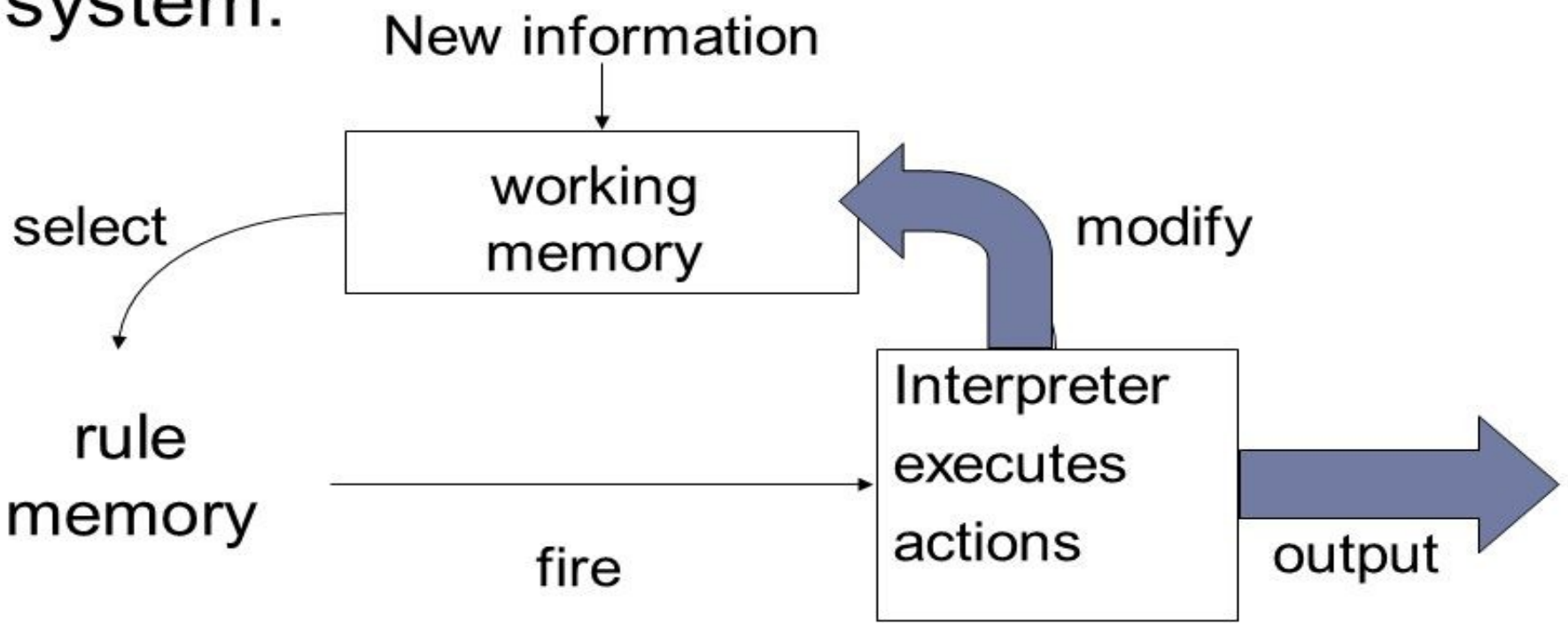
b. One or more knowledge databases that contain whatever information is relevant for the given problem. Some parts of the database may be permanent, while others may temporary and only exist during the solution of the current problem. The information in the databases may be structured in any appropriate manner.

c. A control strategy that determines the order in which the rules are applied to the database, and provides a way of resolving any conflicts that can arise when several rules match at once.

d. A rule applier which is the computational system that implements the control strategy and applies the rules.

Reasoning with production rules

- Architecture of a typical production system:



Reference - Negnevitsky, Artificial Intelligence: A guide to intelligent systems, 2nd Edition, Addison Wesley

Features of Production system

Some of the main features of production system are:

Expressiveness and intuitiveness: In real world, many times situation comes like “if this happen-you will do that”, “if this is so-then this should happen” and many more. The production rules essentially tell us what to do in a given situation.

Simplicity: The structure of each sentence in a production system is unique and uniform as they use “IF-THEN” structure. This structure provides simplicity in knowledge representation. This feature of production system improves the readability of production rules.

Modularity: This means production rule code the knowledge available in discrete pieces. Information can be treated as a collection of independent facts which may be added or deleted from the system with essentially no deleterious side effects.

Modifiability: This means the facility of modifying rules. It allows the development of production rules in a skeletal form first and then it is accurate to suit a specific application.

Knowledge intensive: The knowledge base of production system stores pure knowledge. This part does not contain any type of control or programming information. Each production rule is normally written as an English sentence; the problem of semantics is solved by the very structure of the representation.

Production system characteristics

- Monotonic: execution of a rule never prevents the execution of other applicable rules
- Non-monotonic: which is not Monotonic.
- Partially commutative: if application of a particular sequence of rules transforms state x into state y, then any permutation of those rules that are allowable, transforms x into y
- Commutative: monotonic + partially commutative

Problems and production system

	monotonic	Non-monotonic
Partially commutative	**Theorem proving**	**Robot navigation**
Non-partially commutative	**Chemical synthesis**	**Backgammon** a board game in which two players move their pieces around twenty-four triangular points according to the throw of dice, the winner being the first to remove all their pieces from the board.

Problems and production system

	monotonic	Non-monotonic
Partially commutative	**Ignorable**	**Recoverable**
Non-partially commutative	**Irrecoverable**	**Irrecoverable** **Unpredictable**

Sample problems - 8-puzzle

The 8-puzzle is a small single board player game:

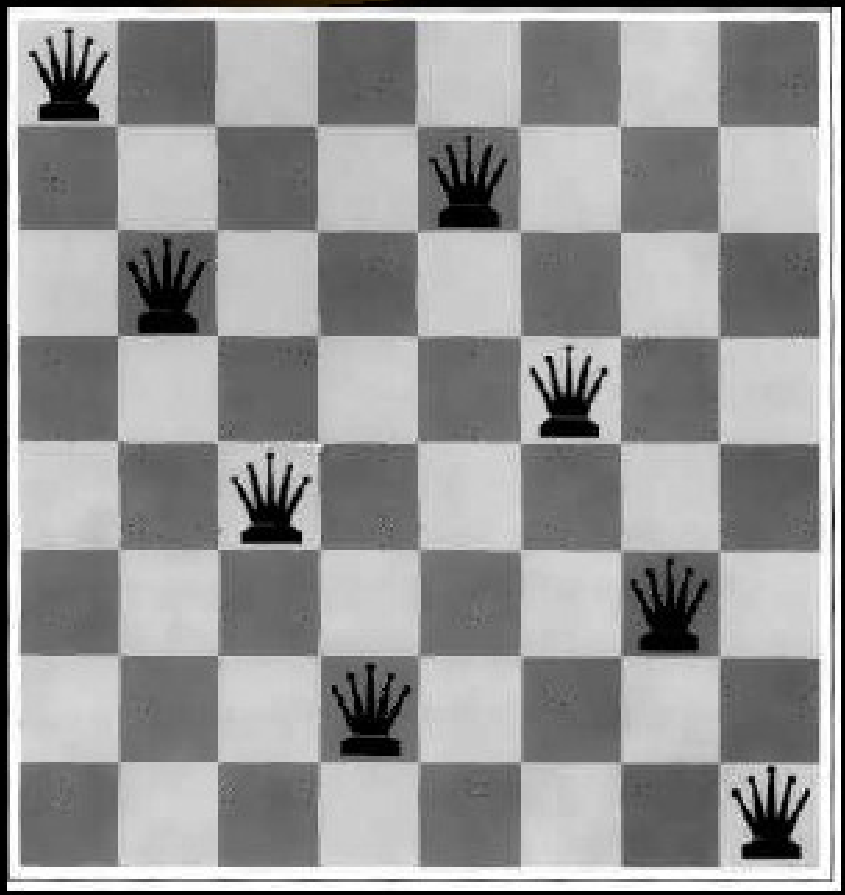
- Tiles are numbered 1 through 8 and one blank space on a 3 x 3 board.
- A 15-puzzle, using a 4 x 4 board, is commonly sold as a child's puzzle.
- Possible moves of the puzzle are made by sliding an adjacent tile into the position occupied by the blank space, this will exchanging the positions of the tile and blank space.
- Only tiles that are horizontally or vertically adjacent (not diagonally adjacent) may be moved into the blank space.

Sample problems - 8-puzzle

- **Goal test:** have the tiles in ascending order.
- **Path cost:** each move is a cost of one.
- **States:** the location of the tiles + blank in the n x n matrix.
- **Operators:** blank moves left, right, up or down.

Sample problems - 8 queens

- In this problem,we need to place eight queens on the chess board so that they do not check each other. This problem is probably as old as the chess game itself, and thus its origin is not known, but it is known that Gauss studied this problem.

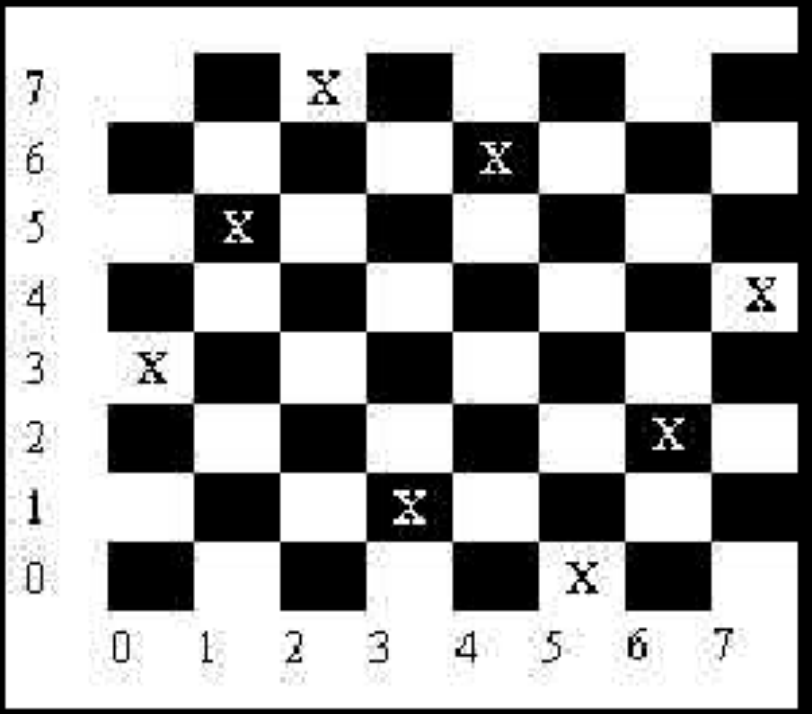


Sample problems - 8 queens

- **Goal test:** eight queens on board, none attacked
- **Path cost:** zero.
- **States:** any arrangement of zero to eight queens on board.
- **Operators:** add a queen to any square

Sample problems - 8 queens

- If we want to find a single solution, it is not hard. If we want to find all possible solutions, the problem becomes increasingly difficult and the backtrack method is the only known method. For 8-queen, we have 96 solutions. If we exclude symmetry, there are 12 solutions.



Sample problems - Cryptarithmetic

- In 1924 Henry Dudeney published a popular number puzzle of the type known as a cryptarithm, in which letters are replaced with numbers.
- Dudeney's puzzle reads: SEND + MORE = MONEY.
- Cryptarithms are solved by deducing numerical values from the mathematical relationships indicated by the letter arrangements (i.e.) . S=9, E=5, N=6, M=1, O=0,….
- The only solution to Dudeney's problem: 9567 + 1085 = 10,652.

Sample problems - Cryptarithmetic

- **Goal test:** puzzle contains only digits and represents a correct sum.
- **Path cost:** zero. All solutions equally valid
- **States:** a cryptarithmetic puzzle with some letters replaced by digits.
- **Operators:** replace all occurrences of a letter with a digit not already appearing in the puzzle.

Issues in design of search problem

Each search process can be considered to be a tree traversal. The object of the search is to find a path from the initial state to a goal state using a tree. The number of nodes generated might be huge; and in practice many of the nodes would not be needed. The secret of a good search routine is to generate only those nodes that are likely to be useful, rather than having a precise tree. The rules are used to represent the tree implicitly and only to create nodes explicitly if they are actually to be of use. The following issues arise when searching:

- The tree can be searched forward from the initial node to the goal state or backwards from the goal state to the initial state.
- To select applicable rules, it is critical to have an efficient procedure for matching rules against states.
- How to represent each node of the search process? This is the knowledge representation problem or the frame problem. In games, an array suffices; in other problems, more complex data structures are needed.

Internal Representation

- In order to act intelligently, a computer must have the knowledge about the domain of interest.
- Knowledge is the body of facts and principles gathered or the act, fact, or state of knowing.
- This knowledge needs to be presented in a form, which is understood by the machine.
- This unique format is called internal representation.
- Thus plain English sentences could be translated into an internal representation and they could be used to answer based on the given sentences.

Properties of Internal Representation

- Internal representation must remove all referential ambiguity.
- Referential ambiguity is the ambiguity about what the sentence refers to.
- Eg: ‘A said that B was not well. He must be lying.’
- Who does ‘he ‘ refers to...?.

Properties of Internal Representation

- Internal representation should avoid word-sense ambiguity.
- Word-sense ambiguity arise because of multiple meaning of words.
- Eg:
  - ‘A caught a pen.
  - A caught a train.
  - A caught fever.’

Properties of Internal Representation

- Internal representation must explicitly mention functional structure.
- Functional structure is the word order used in the language to express an idea.
- Eg: ‘A killed B. B was killed by A.’
- Thus internal representation may not use the order of the original sentence.
- Internal representation should be able to handle complex sentence without losing meaning attached with it.