

* What is artificial intelligence?

AI is the study of how to make computers do things which at the moment, people do better.

* AI problems?

It appeared initially that computers could perform well at those tasks simply by being fast at exploring a large no. of sol. paths and then selecting the best.

It was thought that this process req. very little knowledge and could \therefore be programmed easily.

AI focuses on:

COMMONSENSE REASONING: It includes reasoning about physical obj and their relationships to each other. eg. obj can be only at 1 place at a time.

NATURAL LANGUAGE UNDERSTANDING: The prob. of understanding spoken lang is a perceptual prob. and is hard to solve. But suppose we simplify it by restricting it to written lang. This prob. is ref. to 'NLU'.

The prob. areas where AI is flourishing are the domains that req. only spe. expertise without assistance of commonsense knowledge.

- MUNDANE TASKS: It includes
 - Perception
 - Natural lang
 - Commonsense reasoning
 - Robot control
- FORMAL TASKS: It includes
 - Games
 - Mathematics
- EXPERT TASKS: It includes
 - Engineering
 - Scientific analysis
 - Medical diagnosis
 - Financial analysis.

* What is an AI technique?

One of the few hard & fast results to come out of AI research is that "Intelligence requires knowledge".

Some properties of knowledge are:

- It is voluminous
- It is hard to characterize
- It is constantly changing
- It differs from data being organised to it being used.

Knowledge should be represented in such a way that:

- It captures generalizations.
- It can be understood by people who must provide it.
- It can easily be modified.
- It can be used in many great situations.
- It narrows the range of possibilities that must be considered.

* TIC-TAC-TOE :

board :

prog (1)

1	2	3
4	5	6
7	8	9

An element contains value: 0 is corresponding is blank

1 if " filled wt x.

2 if " " " 0.

Algo:

- ① view vector board, as a binary no. & convert it to decimal.
- ② Use that no. as index into mounetable and access the stored vector.
- ③ Set board equal to the vector.

comments:

- This prog. is v efficient in terms of time.
- It takes a lot of space to store mounetable.
- If we want extend the game, we will have to start from scratch.
- All entries in mounetable should be specified.

prog (2)

board :

1	2	3
4	5	6
7	8	9

we store : 2 indicating blank

3 indicating x

5 " 0

Turn: An integer indicating which move of the game is about to be played.

1 → first move

9 → last move.

Algo:

① Return 5 if center square is blank. Otherwise the funcⁿ returns any blank noncorner square (2, 4, 6 or 8)

② Return 0 if player P cannot win on next move. otherwise, it returns the no. of the sq. that constitutes the winning move.

③ go(n) → make a move in sq. 'n'.

The strategy for each turn is as follows:

TURN = 1 go(1).

TURN = 2 If Board[5] is blank, go(5) else go(1).

TURN = 3 If " [9] " , go(9) " " .

TURN = 4 If posswin(x) ≠ 0 , go(posswin(x)) else go(Make2)

TURN = 5 " " " , go(posswin(x)) else if

posswin(0) is not 0 go(posswin(0))

else if Board[7] blank → go(7) else go(

TURN = 6 If posswin(0) ≠ 0 , go(posswin(0)) else if

posswin(0) ≠ 0 , go(posswin(x)) , else go(Make2)

TURN = 7 If posswin(x) ≠ 0 , go(posswin(x)) else if

posswin(0) ≠ 0 , go(posswin(0)) , else anywhere blank

TURN = 8 If posswin(0) is ≠ 0 , go(posswin(0)) , else if

posswin(x) ≠ 0 , go(posswin(x)) , else anywhere blank.

TURN = 9 Same as TURN = 7

comments:

This prog. is not quite efficient in terms of time.

Prog (3)

Board: A str containing a 9 element vector rep. the board.

A list of board posⁿ that could result from next move.

Algorithm:

To decide whether which set of board posⁿ is best, do:

- ① see if it is a win, call the best by giving it the highest poss. rating.
- ② otherwise, consider all moves the opponent could make next.
- ③ The best move is then the one with highest rating.

This algo is called MINIMAX PROCEDURE.

Comments:

This prog. will req. more time but it could be extended to handle games more complicated than tic-tac-toe.