UNIT – 1
ARTIFICIAL INTELLIGENCE CS6659

Means End Analysis

Means-Ends Analysis (MEA)

- It centers around the detection of difference between current and goal states.
- Once such a difference is isolated, an operator that can reduce the difference must be found.
- The action is performed on the current state to produce a new state.
- The process is recursively applied to this new state and the goal state
- It is quite possible that chosen operator can not be applied to the current state immediately

EXAMPLE

- Assume S and G are start and goal states

S	**B______C**	**G**
Start	**Operator**	**Goal**

- By solving B to C, difference between S, B and C, G is reduced.
- Order in which the differences are considered is critical.
- Important that significant difference be reduced before less significant otherwise great deal of effort may be wasted.
- This method is not adequate for solving complex problems, since working on one difference may interfere with the plan of reducing another.

CONT…

- Reduce the difference between S & B and similarly between C & G.
- MEA also relies on a set of rules just like in other problem solving techniques.
- These rules are usually not represented with complete state description on each side.
- Instead they are represented as
  - left side that describes preconditions and
  - right side that describes those aspects of the problem state that will be changed by application of the rule.

EXAMPLE: HOUSE HOLD ROBOT

- Find a sequence of actions robot performs
  - **Move desk with two things on it from one location S to another G.**
- Operators are: PUSH, CARRY, WALK, PickUp, PutDown and PLACE.
- Main difference :
  - Change of location of desk from initial position to goal position
- Here objects on top must also be moved.
- Data structure called 'Difference Table' indexes the rules by the differences that they can be used to reduce.

CONT…

Operator	Pre-conditions	Result
PUSH(obj, loc)	At(robot, obj) Λ Large(obj) Λ Clear(obj) Λ AE	At(obj, loc) Λ At(robot, loc)
CARRY(obj, loc)	At(robot, obj) Λ Small(obj)	At(obj, loc) Λ At(robot, loc)
WALK(loc)	None	At(robot, loc)
PU(obj)	At(robot, obj)	Hold(obj)
PD(obj)	Hold(obj)	~Hold(obj)
PLACE(Obj1, obj2)	At(robot, obj2) Λ Hold(obj1)	On(obj1, obj2)

Difference Table

	PUSH	CARRY	WALK	PU	PD	PLACE
Move object	*	*				
Move robot			*			
Clear object				*		
Get obj1 on obj2						*
Get arm empty					*	*
Hold object				*		

THANK YOU