CS-TR-77-5

The Coordination of Multiple Goal Satisfaction

Arthur M. Farley
University of Oregon
Computer Science Dept.

In a problem-solving system which knows how to solve most problems, there remains the task of efficiently executing several solutions concurrently. The system proposed has the ability to take advantage of common states between strategies while avoiding certain forms of conflict. The system is demonstrated on a class of block-pile problems.

THE COORDINATION OF MULTIPLE GOAL SATISFACTION

Arthur M. Farley
University of Oregon

I The Model

In the well-developed problem solving system, much goal satisfaction activity will not involve the use of heuristic methods. Rather, a problem will activate a known, general strategy which, if successfully executed, guarantees goal satisfaction. To be applicable in a variety of contexts, such a strategy must be specified in a high level abstraction space. Solutions to tactical details, necessary to bind a strategy to an actual situation, will be based upon similar strategies at lower abstraction levels. The problem solving system will be expected to operate in situations where several, non-independent goals await satisfaction. To function effectively in such situations, the system must be able to coordinate its execution of the independently generated solution strategies.

There are two general approaches to strategy coordination. Both utilize recognized interactions between the current situation and states of the pending strategies. The first approach would combine the separate solution strategies into one, overall plan (Sacerdoti, 1975). The other approach would postpone coordination decisions, making appropriate action selections during execution of the strategies. This second approach has the advantage that new strategies can become pending at any time without replanning costs. Also, the system can take advantage of interactions between a current situation and states of the pending strategies which may not be foreseeable at the time or abstraction level of prior, overall planning. The aims of the research reported here are to define and solve problems associated with multiple goal satisfaction and to develop formalisms which facilitate effective execution coordination. What follows is *system can select an operator leading from a realized state and then complete the cycle by specifying and executing that operator. Whenever a goal state is realized, its strategy is removed from the pending set.*

However, the solution of a problem may require that a conjunctive set of goal states be satisfied by a current situation. Execution coordination of the associated strategies is then more complex. A state is classified as critical if it lies on all paths from leaf states to goal state of a pending, relevant substrategy. A strategy of a conjunctive set is classified as ungrounded if its goal state denies realization of a critical state of another strategy in the set. The goal state of that other strategy should be realized prior to the goal state of the ungrounded strategy. When a goal state of a conjunctive set is realized, strategies in the conjunctive set which are ungrounded solely due to interaction with the completed strategy become grounded. Only realized goal states of grounded strategies are considered solved. Grounded strategies not in the conjunctive set which have a critical state denying the newly realized goal state become ungrounded. When all goal states of the conjunctive set are realized, such ungrounded strategies become grounded again. During each cycle the system selects an operator leading from a realized state of a grounded strategy to a state not denying a realized goal state of a pending, conjunctive set. Criteria for operator selection include cost estimates and goal state priorities.

REFERENCES

Sacerdoti, E. "The non-linear nature of plans" Proc. IJCAI4, 1975, p. 206-214.

Figure II.1

```
GRASP(bananas)    -  PRE:  AT(monkey, ON(box))
                           AT(box, LOC(bananas))
                     ADD:  HAVE(bananas)

CLIMB-ON(box)     -  PRE:  AT(monkey, LOC(box))
                     ADD:  AT(monkey,  ON(box))
                     DEL:  AT(monkey, LOC(box))

CLIMB-DOWN(box)   -  PRE:  AT(monkey,  ON(box))
                     ADD:  AT(monkey, LOC(box))
                     DEL:  AT(monkey,  ON(box))

PUSH(box, loc)    -  PRE:  AT(monkey, LOC(box))
                              (box is at old-loc, not a precondition)
                     ADD:  AT(box, loc)
                           AT(monkey, loc)
                     DEL:  AT(monkey, old-loc)
                           AT(box, old-loc)

WALK(loc)         -     (monkey is at old-loc, not a precondition)
                     ADD:  AT(monkey, loc)
                     DEL:  AT(monkey, old-loc)
```

The assumed agent of all operators is the monkey.
The first argument of an AT proposition is its object argument; the second is its location argument.

leaf state as any state with an empty SSO component is a leaf state. The SSN component of each state in this strategy is assured to be empty for this problem class.

III. Block Piles

This section presents an example of the action selection model at work coordinating the realization of a conjunctive set of goal states. The model is demonstrated in the context of a class of block pile problems, specified as follows:

Each problem involves nine blocks, labelled A-I, arranged in three piles, labelled 1-3. Blocks can be rearranged by moving the top block of any pile to the top of one of the other two piles. Given an initial arrangement, the goal is to arrange them as illustrated below.

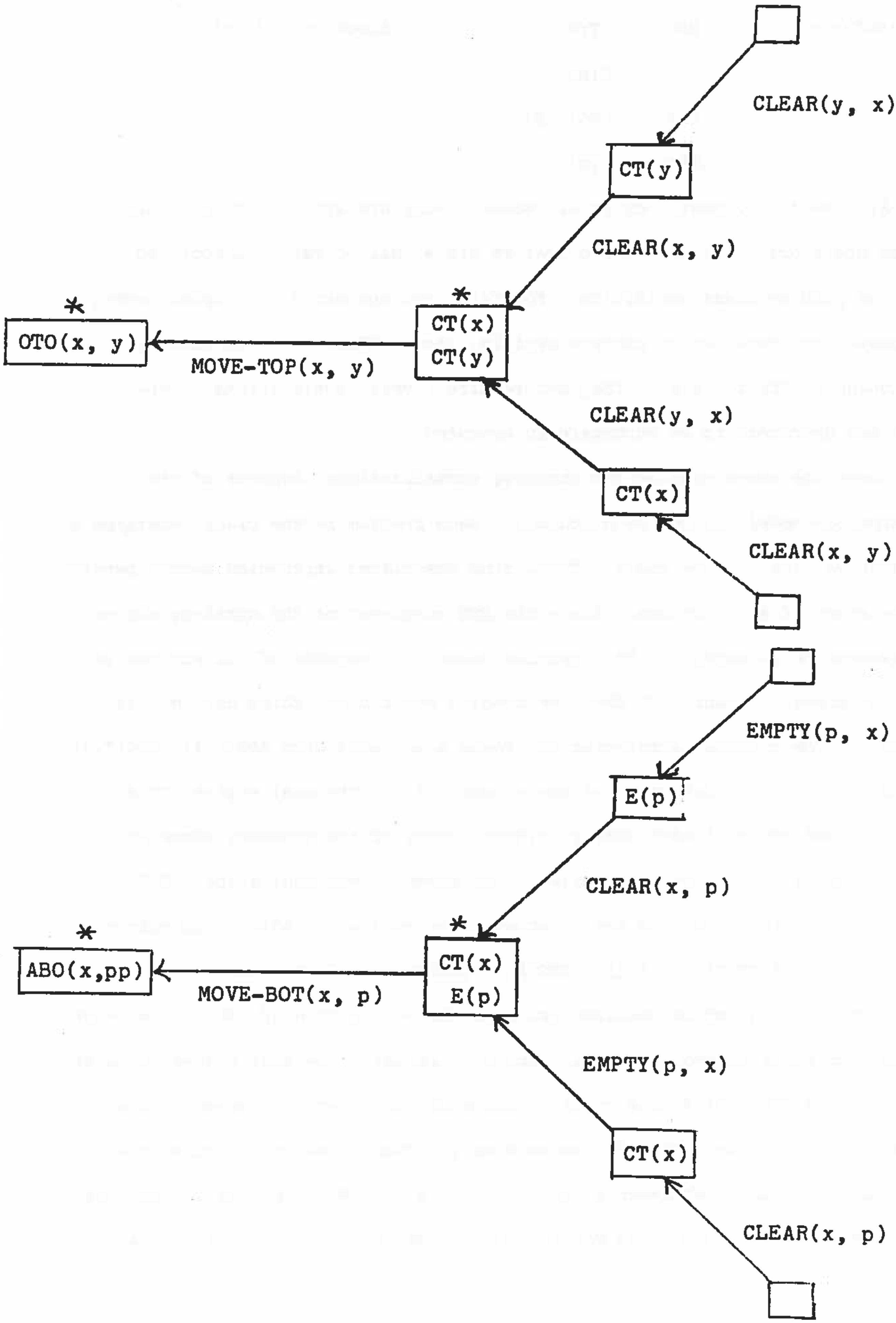
A	D	G
B	E	H
C	F	I
===	===	===
1	2	3

Two types of propositions are necessary aspects of a current situation representing a state occurring in the solution of such problems:

ON-TOP-OF :	OTO(x,y)	-	meaning block x is on top of block y;
AT-BOTTOM-OF:	ABO(x,p)	-	meaning block x is at the bottom of pile p.

The current situation exemplifying the goal arrangement of blocks in these problems therefore consists of nine propositions, as follows:

FIGURE III.1


CLEAR(y, x)
CT(y)
CLEAR(x, y)
*
OTO(x, y)
MOVE-TOP(x, y)
*
CT(x)
CT(y)
CLEAR(y, x)
CT(x)
CLEAR(x, y)
EMPTY(p, x)
E(p)
CLEAR(x, p)
*
ABO(x,pp)
MOVE-BOT(x, p)
*
CT(x)
E(p)
EMPTY(p, x)
CT(x)
CLEAR(x, p)

may become grounded at that time.

An action must be selected from a grounded strategy. The action will be an operator leading from a realized state of a grounded strategy. Let us assume that operator selection is based upon the following, reasonable criteria. An operator is selected which leaves the least number of operator applications before goal state realization. In the strategies of this problem, an operator may leave 0, 1, or 2 operator applications. Of those operators leaving the least operator applications, an operator requiring the least blocks to be moved is selected. Thus, a global and a local measure of future effort are minimized at each point of operator selection.

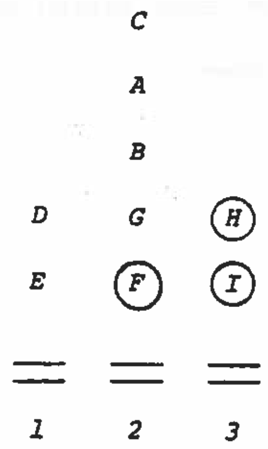
Finally, the model can be demonstrated. Consider the following initial problem state:

I	B	A
D	G	C
E	F	G
1	2	3

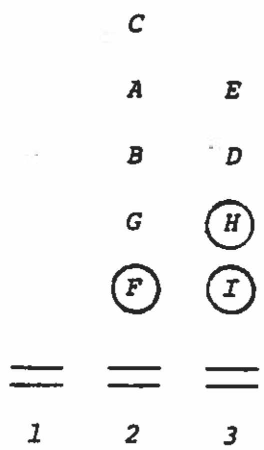
Of the three grounded strategies, the goal state of one, ABO(F,2), is realized. Thus it is considered solved, allowing the strategy associated with OTO(E,F) to become grounded. Note that OTO(D,E) is realized by the initial state, but is not considered solved. Only goal states of grounded strategies can be considered solved. Whenever a block reaches its final resting place, it will be circled, as below.

I	B	A
D	G	C
E	(F)	4
1	2	3

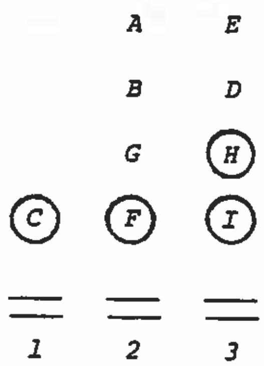
I has a clear top. Therefore the operator which would empty pile 3 is selected. Note that the applicable operators of the other two grounded strategies leave 2



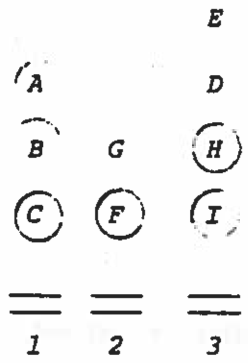
The strategy associated with OTO(G,H) becomes grounded. Now two applicable operators leave 1 operator application. One associated with ABO(C,1) and the other with OTO(G,H). The operator to empty pile 1 for ABO(C,1) is selected as it requires only 2 blocks to be moved, while 3 blocks must be moved to clear the top of G. The current situation becomes



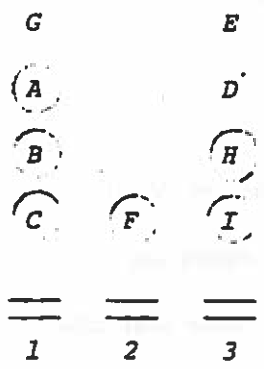
The operator to move C to the bottom of pile 1 is then selected, being the only one leaving 0 operator applications. The current situation becomes



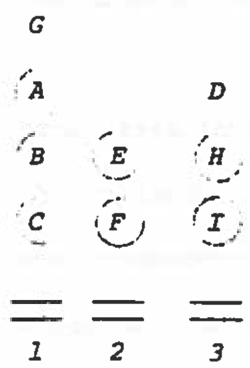
situation becomes



Now the strategies associated with two goal states, OTO(G,H) and OTO(E,F) remain as grounded. Both G and E have clear tops. The operator to clear the top of F is selected as it requires only one block to be moved. The current situation becomes



The operator to realize the goal state OTO(E,F) is then selected. The current situation becomes



The strategy associated with OTO(D,E) becomes grounded. The operator to realize this goal state is selected. The current situation becomes