INTERACTING GOALS AND THEIR USE

Austin Tate
Machine Intelligence Research Unit
University of Edinburgh
Edinburgh, Scotland

Abstract

Current means-end analysis problem solvers are not capable of solving problems which have certain kinds of goal interaction. The goal interaction problem is described and an example given on a block stacking task. Finally, mention is made of a method of using the information gained by the discovery of goal interactions to guide the search for a problem solution.

Interacting Goals

A problem is given to a means-end analysis based problem solver, such as STRIPS (Fikes & Nilsson, 1971) and the planning part of the HACKER (Sussman, 1973) system, as a conjunction of goals

e.g. (G1 & G2)

which must be true for the problem to be solved. Since the individual goals are solved sequentially, they must, once achieved, hold together for a period of time. The time for which an achieved goal must remain true will be called the goal's "holding period". I will illustrate this as in figure

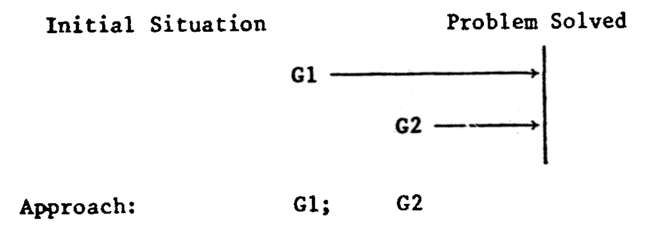


FIGURE 1

The horizontal dimension of this "Holding Period" diagram represents time during which actions will be applied in a final plan to achieve the given goals. Approach should be interpreted as: if G1 not true achieve it using some operator sequence, then do likewise for G2.

STRIPS assumes, in the absence of other information, that it can achieve the individual goals by relevant plan sequences, say, in the order in which the goals are given (Sussman calls this a linear assumption). Thus, as shown in figure 1, it assumes G1 can be solved first by some relevant plan sequence and then that G2 can be solved by a plan sequence following on from the first. If STRIPS can find no way to achieve the goals in the order given, it is capable of reversing the order it has attempted to achieve goals, which were initially not true, at the failure level (e.g. at the top level G1 and G2 could be reversed to give an expected holding period diagram as in figure 2).

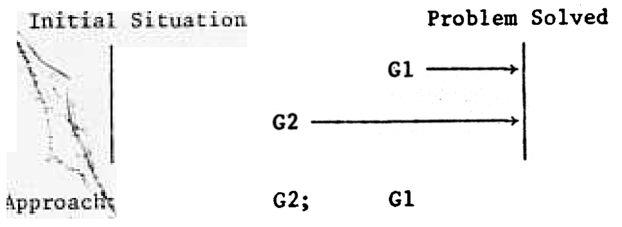


FIGURE 2

STRIPS further assumes that for the goals not already true at the time required, the preconditions, which are required to be true for some operator to be applied to achieve the goal, can all be made true immediately before the time the goal is required to be true. Again, reversals amongst these preconditions can be made on failure backup. Thus, if the preconditions for some operator to achieve a goal G_i are G_{i1} and G_{i2}, then STRIPS initially assumes an approach as in figure 3 can be taken.

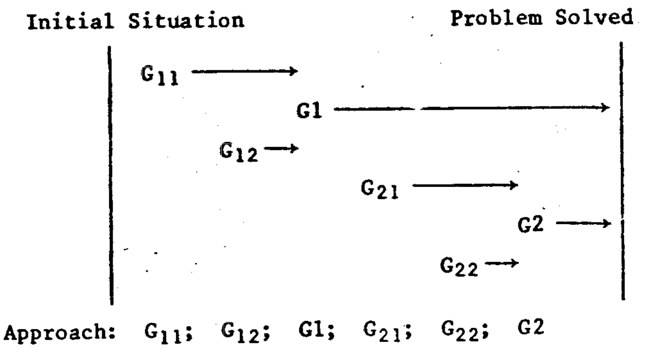


FIGURE 3

Note that the holding period diagram represents the goals to be worked upon for some chosen operator sequence. There is really a 3rd dimension to the diagram representing different choices of operators.

Reversals allow certain other orderings of these goals to be attempted. However, limiting reversals to goals at a particular level of the search tree hierarchy means that STRIPS (these arguments also apply to HACKER) can only tackle certain problems. Specifically, those in which interactions between top level goals can be avoided by suitable ordering of the goals and the choice of suitable operator sequences.

Since STRIPS and HACKER also allow attempts to achieve goals to be repeated if interactions have occurred, they can also handle those problems in which the interactions leave the world in some situation from which the interacted goals can be re-achieved. STRIPS will often produce longer than necessary solutions if it repeats attempts to achieve goals.

Even for very simple worlds, such as the blocks world used by Sussman, interactions can occur. To be able to deal with all types of interaction between a set of goals, we could consider the search space as containing approaches with every interleaving of the goals and the subgoals needed to achieve those goals. Thus, a holding period diagram and approach as shown in figure 4 is necessary to resolve some types of interaction.

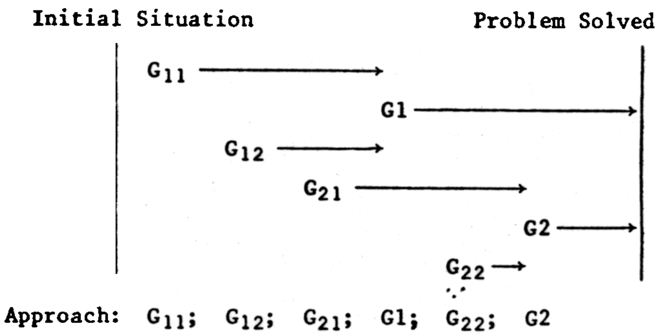


FIGURE 4

The 3-Block Problem

The 3-block problem is an example used by Sussman (1973) in his description of HACKER. It is regarded by HACKER as an Anomalous Situation. The problem is useful as it singles out the interaction difficulty in a simple task.

A world is described by two predicates ON(x,y) and CL(x).

ON(x,y) asserts block x is on top of the (same size) block y.

Note that ON is not transitive.

CL(x) asserts block x has a clear top

There are two operators:-

PUTON(x,y) asserts ON(x,y) and deletes CL(y).

If $\exists$u.ON(x,u) before the application of the operator then assert CL(u) and delete ON(x,u).

It can be applied if CL(x) and CL(y) are true.

ACTCL(x) asserts CL(x).

If $\exists$u.ON(u,x) before the application of the operator then assert CL(u) and delete ON(u,x) and

repeat if $\exists$v.ON(v,u) etc.

(This operator therefore clears all blocks from the top of block x.)

It can always be applied.

Given an initial situation ON(C,A) & CL(C) & CL(B) as shown in figure 5(a) a goal of ON(A,B) & ON(B,C) is given as shown in figure 5(b).

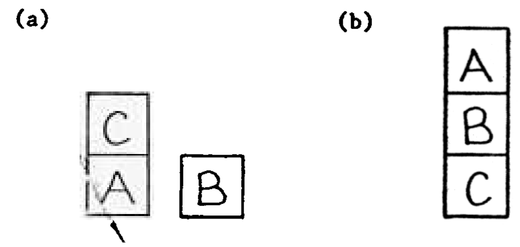


FIGURE 5

STRIPS can tackle (ON(A,B) & ON(B,C)) both of which are not true initially. The goals may, at first, be attempted as shown in the holding period diagram of figure 6.

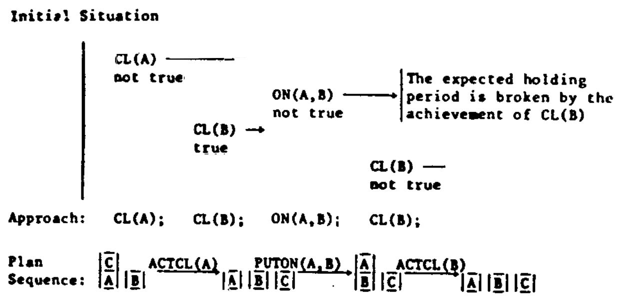


FIGURE 6

The earlier achieved goal (ON(A,B)) does not now hold (its expected holding period is broken), but this is not noticed by STRIPS, and problem solving proceeds as in figure 7.

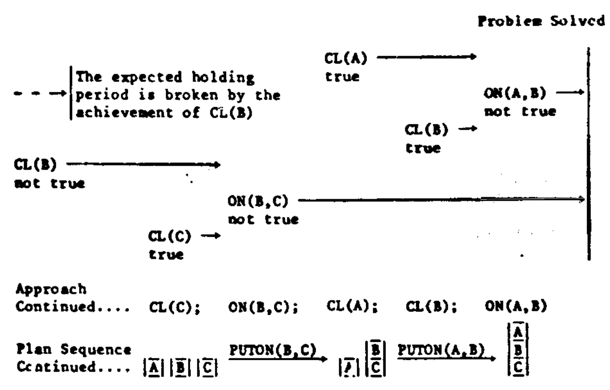


FIGURE 7

So, STRIPS produces the longer than necessary solution:-

ACTCL(A), PUTON(A,B), ACTCL(B), PUTON(B,C), PUTON(A,B).

Attempting the initial goals in the opposite order would make the final solution longer still, though if the interactions in the first ordering produced a situation in which the interacted goals could subsequently not be achieved, this would be attempted on failure backup. STRIPS is incapable of producing a shorter plan for this problem.

HACKER has a mechanism, called Protection, which remembers achieved goals and looks out for actions which violate them. It would notice that the previously achieved goal (ON(A,B)) ceased to hold (as a protection violation) and would try to reverse the order of the top level goals (to ON(B,C) & ON(A,B)) at that time. However, another Protection Violation with the reversed approach will direct the HACKER planner to allow the Protection to be violated, and the result will be the same as STRIPS in this example.

The search space should have included an approach as shown in figure 8.

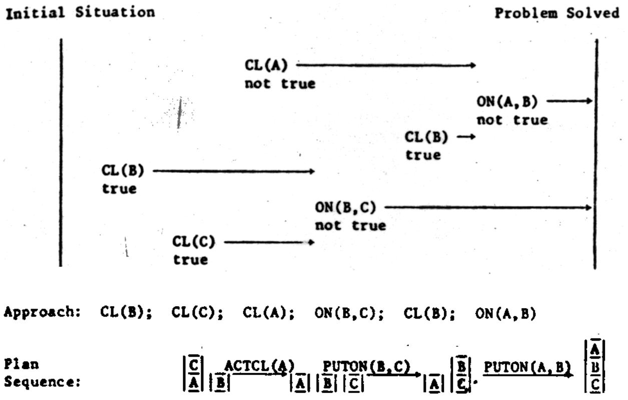


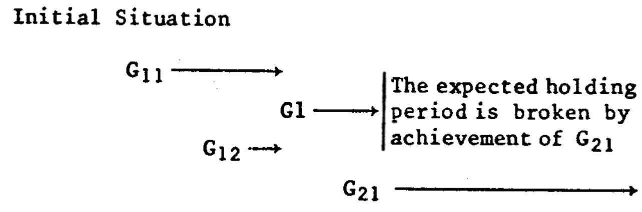
FIGURE 8

STRIPS, by re-achieving the ON(A,B) goal, can solve this problem with a longer than necessary plan because the world produced after interaction is such that the goals can still be achieved. A problem I have been considering - the Keys & Boxes problem (Michie, 1974) - has interactions which would preclude a STRIPS-like problem solver from finding any solution. Interactions occur in other problems, a simple example being the problem of swapping the values of two registers of a computer.

Using goal interactions to suggest new approaches to tackling a problem

Current means-end analysis problem solvers are not capable of solving problems which have certain kinds of goal interaction. Also, with the exception of some systems at MIT (e.g. HACKER), they do not use interactions amongst goals to guide the search for a solution. I mentioned earlier that all interleavings of goals, and the subgoals needed to achieve those goals, should have the potential of being considered. Generally, only very few of the possible interleavings need be considered. An assumption, such as is made by many existing problem solvers, that goals can be achieved in the order given without interaction (linearily) is, however, a very powerful heuristic. My own work in problem solving (Tate, 1974) is based upon the powerful heuristics used in STRIPS and other problem solvers, but I am anxious not to let these assumptions rule the type of problems which can be dealt with. Proven contradictions of these assumptions during problem solving can direct the search to consider appropriate interleavings of plan parts to remove interactions.

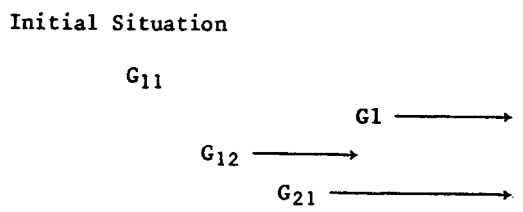
The information gained from the discovery of an interaction can be used to suggest appropriate continuations. As an example, the interactions discovered during attempts to solve the goals G1 & G2 linearily can lead us to the point, in figure 9,



Approach: G_{11}; G_{12}; G1; G_{21};

FIGURE 9

where the expected holding period for G1 is broken by the achievement of a subgoal G_{21} required for an action to achieve G2. We have tried and found that G1 and G_{21} cannot both hold together when they have been achieved by some operator sequences in the order G1 and then G_{21}. We can either try an approach in which the goals at a higher (here the top) level are reversed to stop the conflicting goals' holding periods overlapping altogether (by reversing G1 and G2) or try to achieve the conflicting goals in the opposite order. It is sufficient to try to achieve the conflicting goals in the other order only once. This can be done whilst still preserving linearity as far as possible by moving the precondition (G_{21}) whose achievement made a previously achieved goal (G1) not hold, immediately in front of the goal as shown in figure 10.



Approach: G_{11}; G_{12}; G_{21}; G1;

FIGURE 10

We shall say that we PROMOTE the precondition. Moving it further back through the goals to be worked upon would still try to achieve the conflicting goals in the opposite order but would risk further possibilities for other intermediate goals to interact with the precondition being promoted. Note that the promoted precondition (G_{21}) may interact with earlier goals and may need to be shifted again due to different interactions. Subgoals intermediate between G2 and G_{21} if they exist may need to be promoted also.

If in both orders the same goals achieved by suitable operator sequences still interact and cannot hold together, the problem cannot be solved by this approach.

The technique described above has been incorporated into a problem solver, INTERPLAN (described in Tate, 1974), which has been applied to a variety of problems. A trace of INTERPLAN on the 3-blocks problem is given in figure 11.

```
: GOAL <<ON A B>> <<ON B C>>;

ENTERING INTERPLAN WITH INITIAL SITUATION 1

** ACHIEVE << ON A B >> IN 1                  attempting <<ON A B>>
** ACHIEVE << CL A >> IN 1                      and then <<ON B C>>
** APPLY << ACTCL A >> TO 1 TO GIVE 2
** APPLY << PUTON A B >> TO 2 TO GIVE 3
** ACHIEVE << ON B C >> IN 3
** ACHIEVE << CL B >> IN 3
** APPLY << ACTCL B >> TO 3 TO GIVE 4
 PROTECTION VIOLATION REORDER
** ACHIEVE << ON B C >> IN 1                  attempting <<ON B C>>
** APPLY << PUTON B C >> TO 1 TO GIVE 5         and then <<ON A B>>
** ACHIEVE << ON A B >> IN 5
** ACHIEVE << CL A >> IN 5
** APPLY << ACTCL A >> TO 5 TO GIVE 6
 PROTECTION VIOLATION PROMOTE
** ACHIEVE << CL A >> IN 1                      attempting <<CL A>>
** APPLY << ACTCL A >> TO 1 TO GIVE 7                then <<ON B C>>
** ACHIEVE << ON B C >> IN 7                    and then <<ON A B>>
** APPLY << PUTON B C >> TO 7 TO GIVE 8
** ACHIEVE << ON A B >> IN 8
** APPLY << PUTON A B >> TO 8 TO GIVE 9

** CPU TIME = 2.109 SECS

NOW
<< ACTCL A >>
<< PUTON B C >>
<< PUTON A B >>

: APPROACH

-1001  << CL A >>          -1001 indicates << CL A >> is a
2  << ON B C >>            precondition for the goal ref.1
1  << ON A B >>
```

FIGURE 11

References

(1)

(3) Sussman, G.J. (1973) A computational model of skill acquisition. MIT Technical Report AI TR-297.

(4) Tate, A. (1974) INTERPLAN: A plan generation system which can deal with interactions between goals. Research Memorandum MIP-R-109. Edinburgh: Machine Intelligence Research Unit.

Acknowledgements

I thank the designers of STRIPS and HACKER for providing their excellent problem solvers on which I can base these comments. Many points have been raised or clarified in discussion with Harry Barrow. Financial support for this work was provided by a Science Research Council Studentship.