INTERACTING GOALS IN PROBLEM SOLVING

AUSTIN TATE JUNE 1974

A problem is given to a means-end analysis driven problem solver, such as STRIPS (1) and the planning part of Sussman's HACKER (2) system, as a conjunction of goals

e.g. (G1 & G2)

These must be true at the end of the problem, and as they are solved sequentially, the goals must hold together for a period of time, as first one, then the other is achieved. The time for which a goal must remain true will be called the goal's "holding period". I will illustrate this as in figure 1.

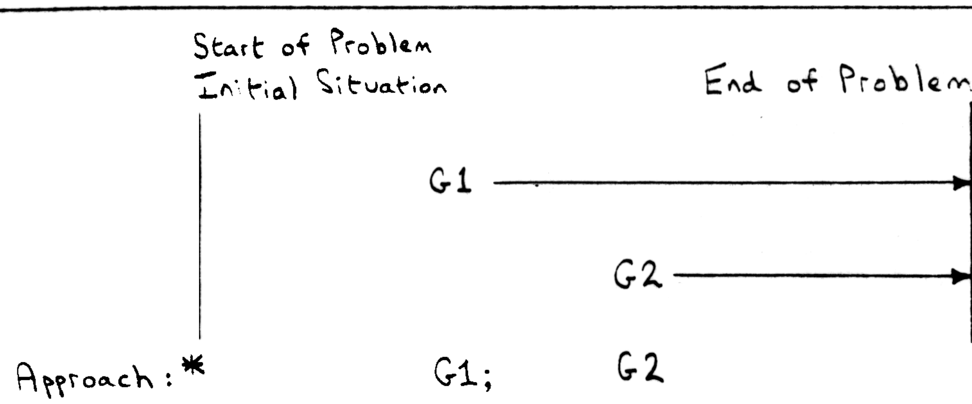


FIGURE 1

The horizontal dimension of this "Holding Period" diagram represents time during which actions will be applied in a final plan to achieve given goals.

STRIPS assumes, in the absence of other information, that it can achieve the goals by plan sequences 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 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

* The approach should be read as: if G1 not true achieve it using some operator sequence, then do likewise for G2.

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).

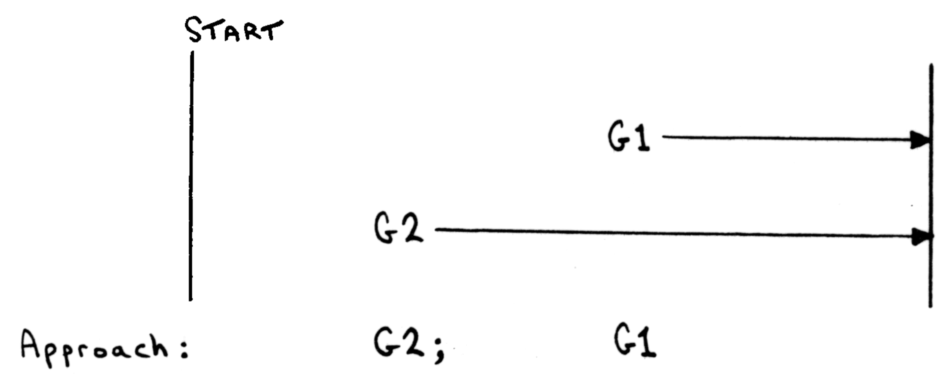


FIGURE 2

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 used 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.

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

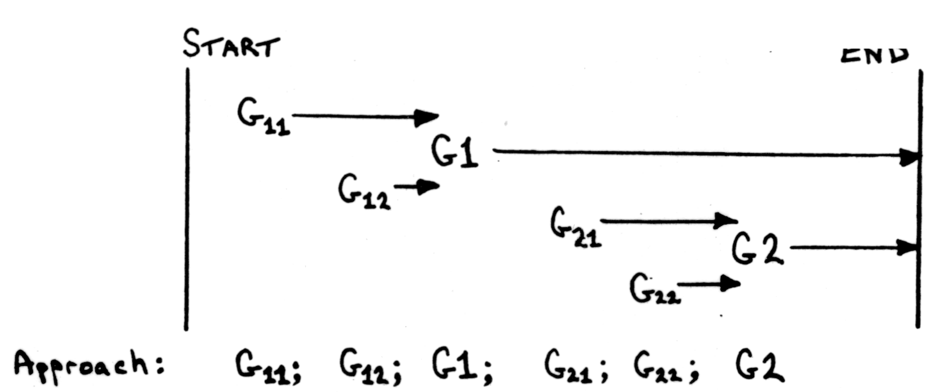


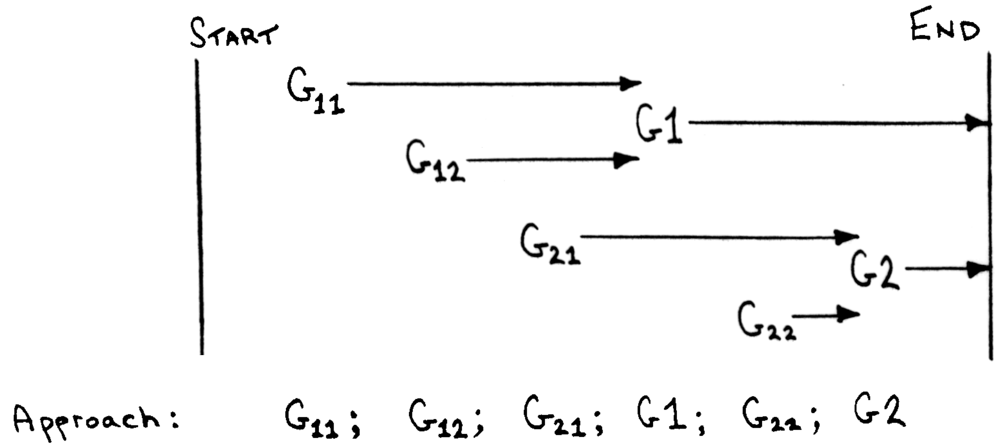
FIGURE 3

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, complex interactions can occur. To be able to deal with all types of interaction between goals, we could consider the search space as containing every interleaving of the goals and subgoals at all hierarchical levels of the search tree. Thus, a holding period diagram and approach as shown in figure 4 is necessary to resolve some types of interaction.

FIGURE 4



EXAMPLE (A simple version of Sussman's blocks world and an example from this is used.)

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.

CL(x) asserts block x has a clear top.

There are two operators:-

PUTON(x,y) deletes any fact ON(x,z) and asserts CL(z) for it. It asserts ON(x,y) and makes CL(y) false. It can be applied if CL(x) & CL(y) is true.

ACTCL(x) asserts CL(x), making false any relations ON(y,x); ON(z,y) etc. All blocks (y) removed are put somewhere in free space and CL(y) is asserted for them.

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).

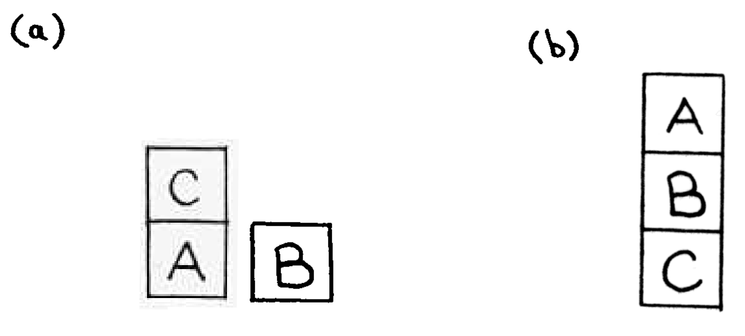


FIGURE 5

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

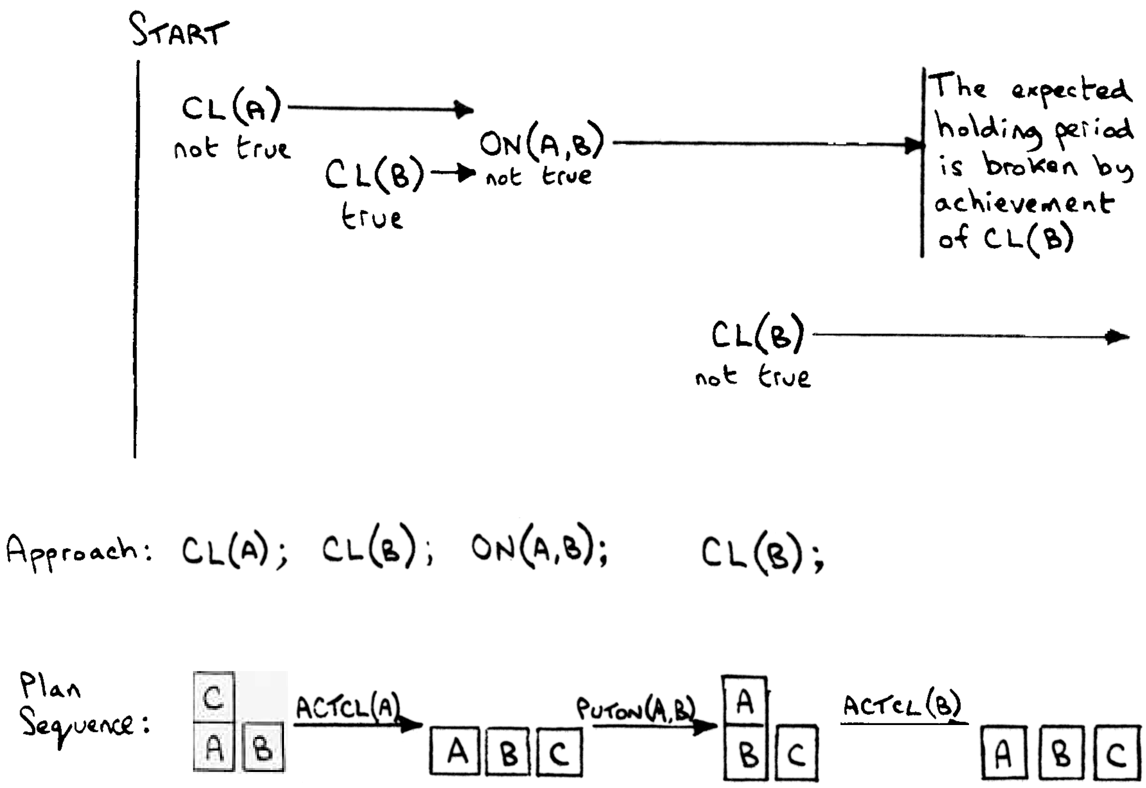
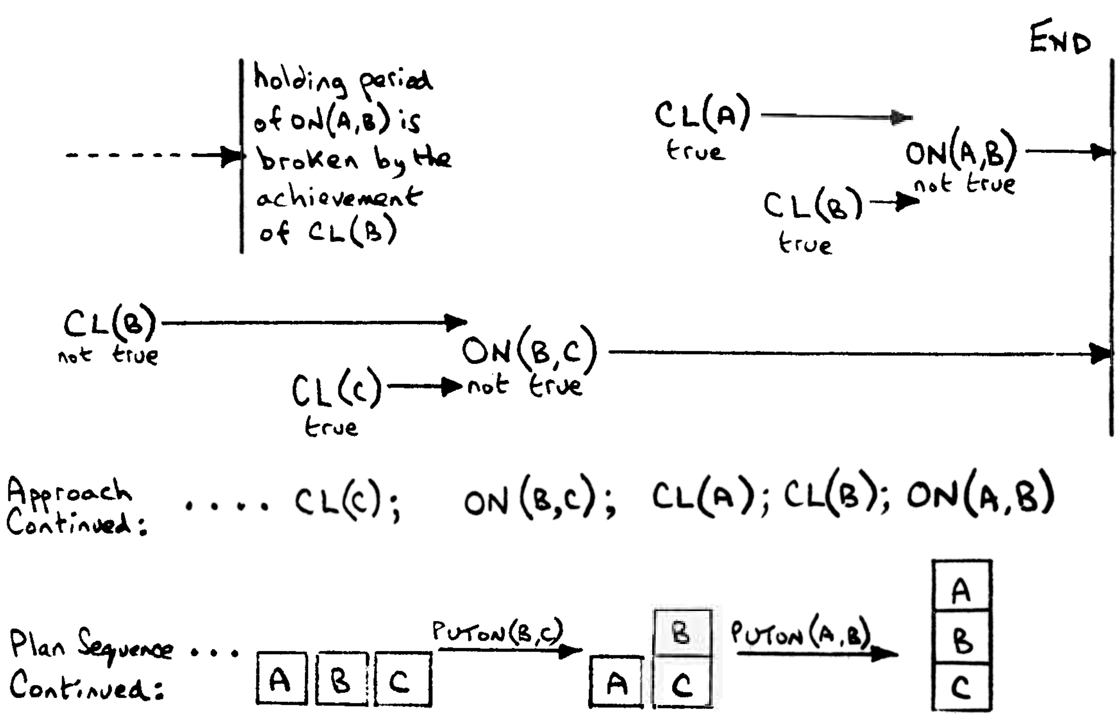


FIGURE 6

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



STRIPS produces the longer than necessary solution:-

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

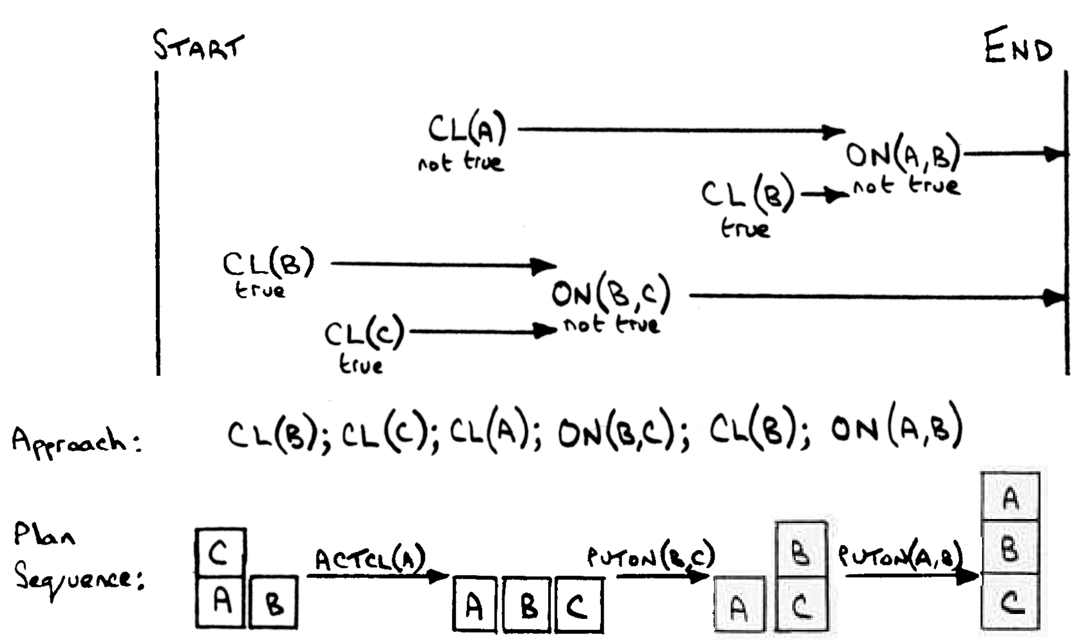
Putting the initial goals in the opposite order would make the final solution longer still, though if the interactions in the first ordering were non-recoverable this would be attempted on failure backup.

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 this reversed attempt will direct the HACKER planner to allow a Protection Violation and the approach will be the same as STRIPS in this example.

The search space should have included an approach as shown in figure 8. This approach is an ordering not allowed by reversals only within the hierarchic levels of the search tree. It would have led to a solution plan:-

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

FIGURE 8



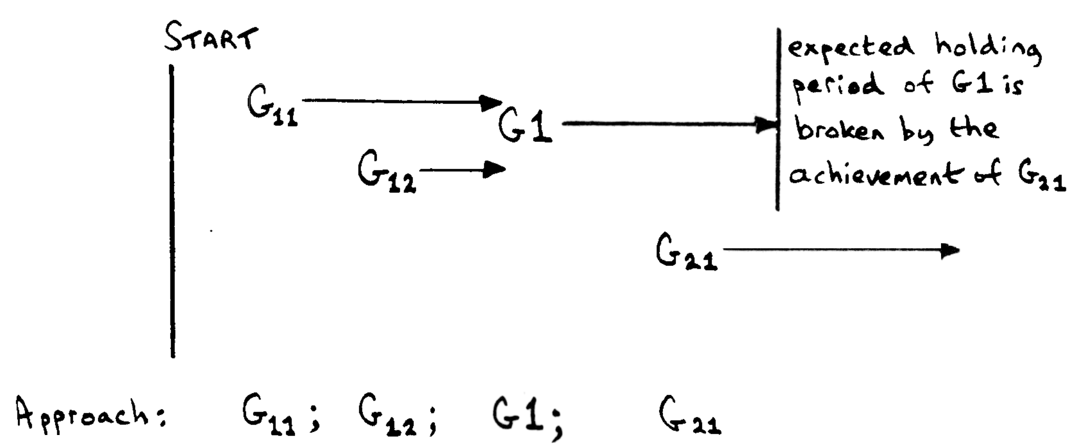
Although STRIPS, using re-achievement, can solve this problem with a longer than necessary plan because the interactions do not destroy some needed information in the world, a problem I have been considering The Keys & Boxes Problem (3) - has interactions which would preclude a STRIPS-like problem solver from finding any solution.

Summary

Current means-end analysis problem solvers are not capable of solving problems which have certain kinds of goal interaction, and (with the exception of some systems at MIT e.g. HACKER) do not use interactions among goals to guide the search for a solution. I mentioned earlier that all interleavings of goals at any level of the hierarchy of the search tree should really be considered. Generally, only very few of the possible interleavings need be considered. An assumption that goals can be achieved in the order given without interaction (linearily) is, however, a very powerful heuristic. My own work in problem solving is based upon the powerful heuristics used in STRIPS and other problem solvers, but I am anxious not to let these assumptions rule the types of world I can deal with. Proven contradictions of these assumptions during problem solving can direct the search to consider interleavings of plan parts to remove interactions.

As an example, the interactions discovered during attempts to solve the goals G1 & G2 linearily lead us to the situation, in 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

FIGURE 9



been achieved by some operator sequences in the order G1 and then G_{21}. We can either try to achieve the conflicting goals in the opposite order, or reverse goals at a higher level to stop their holding periods overlapping altogether (be reversing G1 and G2). It is sufficient to try to achieve the conflicting goals in the other order only once, and this can be done whilst still preserving linearity as far as possible by moving the precondition (G_{21}) which made a previously achieved goal (G1) not hold, immediately in front of the goal as shown in figure 10. Moving it further back through the goals to be worked

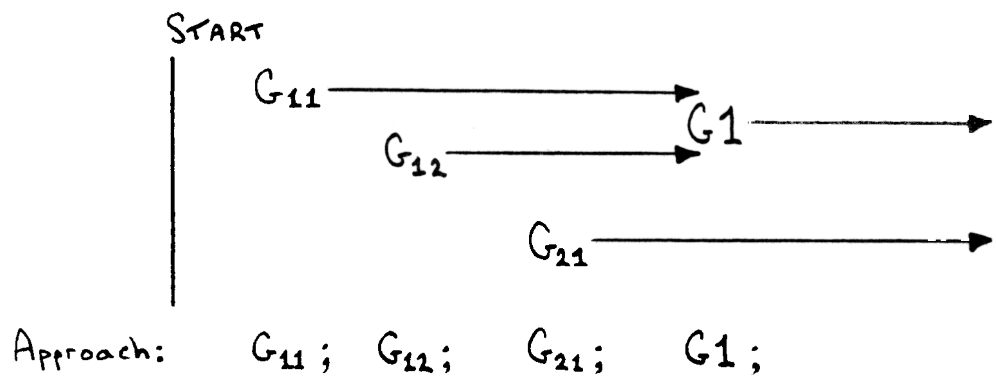


FIGURE 10

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 brought forward.

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

Note that the precondition brought forward (G_{21}) may interact with earlier goals and may need to be shifted again due to different interactions.

References

Fikes, R.E. and Nilsson, N.J. (1971) STRIPS: a new approach to the application of theorem proving to problem solving. Artificial Intelligence, 2, 189-208.

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

(3) Michie, D. (1974) On Machine Intelligence Edinburgh: Edinburgh University Press.

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.

DEPARTMENT OF MACHINE INTELLIGENCE
HOPE PARK SQUARE,
MEADOW LANE