Some Mathematical Structures Underlying Efficient Planning

Aarati Parmar
Department of Computer Science,
Gates Building, 2A wing
Stanford University, Stanford, CA 94305-9020, USA
aarati@cs.stanford.edu

Abstract

We explore *antimatroids*, also known as shelling structures, a construct used to formalize when greedy (local) algorithms are optimal, as well as their relation to the *strong measure of progress* P introduced in (Parmar 2002b). We begin with an example from the map coloring domain to spark the reader's intuitions, and then move towards a more general application of shelling to the strong measure of progress. We also introduce some extensions of shelling to planning on a different level. Macro-operators are another kind of mathematical structure that help give efficient and easy-to-understand plans, but we must be careful how we use them when defining strong measures of progress.

1 Introduction

Current planning technologies often rely on numerical estimates to ascertain which actions make progress towards the goal. (Parmar 2002b) is the first attempt to describe this progress logically in terms of local predicates of the domain. Part of the rationale behind such a "logical measure of progress" is that it identifies subgoals which can be completed immediately, without clobbering other subgoals. Since it is declaratively represented, the logical measure of progress is easily understandable from its formulation, and can be easily modified, compared to numerical estimates. As an example, consider the goal of attending the AAAI Spring Symposium. One has to drive to Stanford, and then go to the lecture hall. A planner using a numeric-based estimate would point the way to the goal by deeming the value of driving to Stanford as higher than say, driving to Berkeley. A logical measure of progress would instead explicitly describe driving to Stanford as a necessary landmark needed to get to the lecture hall.

This paper examines some of the issues related to constructing a logical measure of progress. The main thread is concerned with how to decompose a planning problem into smaller, easier problems, based on the structure of a domain. (Long & Fox 2000; Fox & Long 2001) have already formalized some structure in terms of *generic types*, which represent common sense concepts such as *mobiles* (objects which can move to different locations) and *maps* (routes upon which mobiles can move). These common sense concepts provide some limited but powerful semantics which is used to constrain search whilst planning.

At the other end of the spectrum, (Bylander 1994) gives very general complexity results for propositional STRIPS-style planning problems in terms of the structure of the operators. Recall that a STRIPS style o operator consists of three lists: a precondition list o^p, an add list o^a, and a delete list o^d. The postconditions are the union of o^a and o^d; o^a are the positive postconditions, and o^d the negative ones. Bylander assumes that each list consists only of ground literals, and that the goal is a set of ground literals, while the initial state is a set of atomic formulas. He then shows that finding a plan (or proving none exist) in this kind of domain is polynomial in the size and number of operators when:

1. Each operator has any number of positive preconditions and at most one postcondition, or
2. Each operator has exactly one precondition and there are a bounded number of goal literals, or
3. Each operator has zero preconditions.

Case 1 is polynomial because one can first apply all of the operators with positive postconditions in order to get a maximal superset of the goals true, and then apply the operators with negative postconditions to remove all negative goals. Cases 2 and 3 both are proven using backward search from the goal. 2. works because each goal can be broken down into at most one subgoal, which keeps backward search from the goal polynomial as long as the number of goals is fixed. In case 3, we start with the goal, only considering operators that do not clobber the goals, and then recurse on the reduced problem where the goals are those minus the ones achieved by the operator. When the remaining goals are true in the initial state, we have a plan.

The results in (Bylander 1994) are impressive, despite the restrictions, because they detail polynomial-time algorithms for planning based only on rather coarse structure. One way we differ in our approach is that we take a finer-grained view of this mathematical structure, looking closer at dependencies between goals and operators. Also, we study a more specific kind of [lack of] complexity than computational complexity, that is, when we can identify some notion of progress for a planning domain. However, the existence of this measure of progress will imply some kind of polynomial complexity. It is interesting to note that the rationale behind cases 1 and 3 do use some notions of progress; 1.

first tries to accomplish all positive goals, and then all negative ones, while 3. works by avoiding clobbering, which is the dual notion to making progress.

We explore three topics:

1. What is the underlying mathematical structure that allows us to have a logical measure of progress for some planning domains?

   We begin with a map coloring example in Section 3 as a clear demonstration of planning without search. The example provides not only the mathematical structure, but the intuition surrounding this concept. In Section 4, we define the strong measure of progress introduced in (Parmar 2002b), and examine how our mathematical structures relate to it.

2. Shelling, a key concept underlying greedy progress, may be more generally applicable to planning problems. We provide some indications of this in Section 5.

3. In Section 6, we discuss the utility of adding macro-operators to our theory and how it affects our strong measure of progress.

2 Mathematical Preliminaries

First we provide some basic definitions from graph and set theory, and then introduce antimatroids. Throughout this paper we assume that all sets are finite and non-empty.

Let V be a set. (V, E) is a graph with vertices V and edges $E \subseteq V^2$. If $U \subseteq V$, the notation $(V, E) \setminus U$ refers to the graph with nodes in U, and all edges referring to them, removed from (V, E). We treat cartographic maps as graphs.

Let A be a set. $(A, \mathcal{L})$ is a *language* over the alphabet A if $\mathcal{L}$ contains some subset of the sequences of letters from A: $\mathcal{L} \subseteq A^*$. We use the Greek letters α and β to refer to strings in $\mathcal{L}$. The symbol $\emptyset$ refers to the empty sequence. We use letters x, y, z to refer to symbols of A. $x \in \alpha$ means the symbol x occurs in the string α. $\hat{\alpha}$ is α's *underlying set*: $\hat{\alpha} = \{x \in A \mid x \in \alpha\}$. We distinguish between the string $(x_1 \dots x_k)$ and the set $\{x_1, \dots, x_k\}$. Also, $a <_\alpha b$ means the symbol a occurs before b in string α, assuming no repetitions of either element.

The language $(A, \mathcal{L})$ is *normal* if every symbol in A appears in some word of $\mathcal{L}$. $(A, \mathcal{L})$ is *simple* if no string in $\mathcal{L}$ has a repeated element. A *hereditary* language $(A, \mathcal{L})$ is one that is closed under all prefixes: $\emptyset \in \mathcal{L}$, and $\alpha\beta \in \mathcal{L} \implies \alpha \in \mathcal{L}$.

Greedoids (Korte, Lovász, & Schrader 1991) are the mathematical structures which enable greedy algorithms to reach optimal solutions. They can be divided into two mutually exclusive classes, the *matroids* and the *antimatroids*. We can visualize antimatroids, or *shelling structures* (Korte & Lovász 1981), as structures which can be readily decomposed by removing successive layers until nothing is left.

(Korte, Lovász, & Schrader 1991) show how convex geometries in Euclidean spaces illustrate this notion of shelling. A set X is convex if every line connecting two points in X is contained in X. An *extreme point* x of a set X is one that is not included in the convex hull (the convex closure) of $X \setminus x$. The *shelling* of a set X is the process by which the extreme points of X are progressively removed. One possible shelling is depicted in Figure 1.

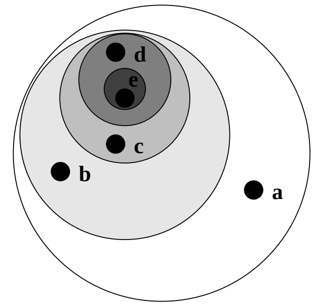


Figure 1: One possible shelling of a set of points. a is not in the convex closure of $\{b, c, d, e\}$, is extreme, and can therefore be removed. Then b can be shelled from $\{c, d, e\}$, and so forth. The resulting shelling sequence for this example is $abcde$. (Keep in mind that there are many other possible shellings for this set.)

Formally, a shelling is represented by a set of strings of X with no repetitions, each sequence recording the order in which extreme points are removed. The set of shelling sequences in a convex geometry is always an antimatroid. Definition 1 below gives a formal definition of these shelling sequences. Note that an antimatroid can also be formulated as a set, explained later below.

Definition 1 (Antimatroids (Korte & Lovász 1981))
Let $(A, \mathcal{L})$ be a language. It is an antimatroid iff:

1. *$(A, \mathcal{L})$ is simple.*
2. *$(A, \mathcal{L})$ is normal.*
3. *$(A, \mathcal{L})$ is hereditary.*
4. *$\alpha, \beta \in \mathcal{L} \land \hat{\alpha} \not\subseteq \hat{\beta} \implies (\exists x \in \alpha)[\beta x \in \mathcal{L}]$*

Each sequence $\alpha \in \mathcal{L}$ describes the order in which elements are shelled from A. Since $(A, \mathcal{L})$ is simple, no element is removed twice. Normality means that every element of A is mentioned in some elimination sequence. The hereditariness closes $\mathcal{L}$ under prefixes, so that it includes all legal partial eliminations. The fourth property, combined with normality and simplicity, guarantee that any partial elimination sequence can be extended to one which removes all elements of A. Lemma 2 proves this:

Lemma 2
Let $(A, \mathcal{L})$ be an antimatroid. Then there exists a string $\alpha \in \mathcal{L}$ containing all elements of A.

Proof: *Assume not. Take any maximal string $\beta \in \mathcal{L}$, whose underlying set is not strictly included in some other string. Consider the symbols $A \setminus \hat{\beta}$ that are missing from β. Since $\mathcal{L}$ is normal, there is another string γ which mentions some of these missing elements. By property 3 in Definition 1, one of the elements in γ can be appended to β to produce another member of $\mathcal{L}$. Since $\mathcal{L}$ is simple, it must be an element not already in β (no repeated elements allowed). But then β is not maximal.* ■

Antimatroids can also be generated by means of a family H_x of sets, known as *alternative precedence structures*. For each $x \in A$, $H_x \subseteq 2^{A \setminus x}$ is any set of subsets of $A \setminus x$. From these sets we can generate a language:

$$\mathcal{L}_H = \{(x_1 \dots x_k) \mid (\forall i \leq k)[x_i \in A \wedge (\exists U \in H_{x_i})[U \subseteq \{x_1, \dots, x_{i-1}\}]]\} \quad (1)$$

The sets $U \in H_{x_i}$ can be thought of as the possible "enablers" for x_i; x_i cannot be shelled away until all of the elements in one of the precedence sets in H_{x_i} have. For the convex shelling example, each set in H_{x_i} corresponds to the elements that need to be removed in order for x_i to be an extreme point of the resulting set. From this one can see how the H_{x_i}s determine the ordering of elements in $\mathcal{L}_H$.

Theorem 1 ties together the above three concepts:

Theorem 1 (Antimatroids (Korte, Lovász, & Schrader 1991)) *Let $(A, \mathcal{L})$ be a simple, normal language. Then the following three statements are equivalent:*

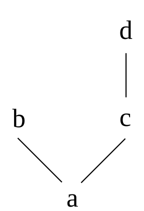
1. *$(A, \mathcal{L})$ is an antimatroid.*
2. *$(A, \mathcal{L})$ is the language of shelling sequences of a convex geometry.*
3. *$(A, \mathcal{L})$ is the language of feasible words of a system of alternative precedences.*

Earlier we mentioned antimatroids can be formulated either in terms of languages, or sets. Any language antimatroid $(A, \mathcal{L})$ can be converted to a set version $(A, \mathcal{F})$ by defining $\mathcal{F} = \{\hat{\alpha} \mid \alpha \in \mathcal{L}\}$. Additionally, an antimatroid $(A, \mathcal{F})$ can be converted to a unique language antimatroid, so the two representations are interchangeable.

Besides convex shellings and alternative precedence structures, one of the most natural ways to generate an antimatroid is through *the shelling of a poset* $(P, \leq)$ (Korte & Lovász 1981), where the $\leq$-minimal elements are repeatedly eliminated from P. The language $\mathcal{L}_{\leq}$ describing these eliminations can be formulated as:

$$\mathcal{L}_{\leq} = \{(x_1 \dots x_k) \mid \{x_1, \dots, x_k\} \text{ is downwards closed under } \leq \text{ and the ordering is compatible with } \leq.\} \quad (2)$$

Figure 2 gives an example poset shelling for the given poset.



Language antimatroid

a	abc	abcd
ab	acb	acbd
ac	acd	acdb

Figure 2: Poset $\leq$ and resulting language antimatroid.

Another way to generate the shelling of a poset is by using alternative precedence structures: fix each H_x to contain the single alternative precedence set, $\{y \in P \setminus x \mid y \leq x\}$.

3 Map Coloring Problem

Map coloring is one domain where planning can be reduced to reasoning about the order in which goals are achieved. The Four Color Theorem (Appel & Haken 1977a; 1977b) guarantees that any map can be colored with at most four colors, and there is already a quadratic algorithm (Robertson *et al.* 1995) for doing so. Our rationale for researching this domain is knowing when and how we can solve the problem *without having to backtrack.* This is the same motivation as that for finding a logical measure of progress for planning.

(McCarthy 1982) reiterates a reduction alluded to by (Kempe 1879) that will never require backtracking: if a country C in a map has three or fewer neighbors, then we can postpone four-coloring C until the other three neighbors have been colored. The original problem of coloring reduces to the same map minus C. In some cases, such as the map of the United States, this process will continue until the map is completely stripped, in which case coloring is done in the reverse order that the countries are eliminated, respecting other countries' colors as required.[1]

We provide a theorem which elucidates the fundamental structure required for maps to be so easily reduced, and shows how they are in fact, antimatroids. The shells removed are the vertices of degree three or less. We generalize to n colors:

Definition 3 (n-reducible maps)
A map (V, E) is n-reducible *if one can repeatedly remove vertices of degree n or less from the graph, until the empty graph is encountered.*

If a map (V, E) is n-reducible then we can color it with $n+1$ colors without ever having to backtrack. We can prove that a map is n-reducible iff its underlying structure is an antimatroid. To do so we first define for each $x \in V$ a set $H_x \subseteq 2^{V \setminus x}$ satisfying the following:

$$H_x = \{U \subseteq V \setminus x \mid x \text{ has degree } n \text{ or less in the graph } (V, E) \setminus U\} \quad (3)$$

The sets in each H_x are just ways we can pare down vertex x in graph (V, E) to having n or fewer neighbors by removing other vertices. This does not involve any reasoning about the order in which nodes are removed, and only involves thinking about removing the nodes which are neighbors of x, as the sets in H_x are closed under supersets.

We define our shelling sequences, $\mathcal{L}_H$ as in (1) using the definition of H_x in (3).

Theorem 2 (n-reducible maps are antimatroids.)
Let (V, E) represent a map. (V, E) is n-reducible iff $(V, \mathcal{L}_H)$ is an antimatroid.

Proof: $\rightarrow$*: If (V, E) is n-reducible, then there is a sequence of shelling vertices so that the empty graph is reached; that is, there is a sequence α which uses all elements in V. If we can show $\alpha \in \mathcal{L}_H$, then we are done,*

[1] We do not address the more complicated reduction for countries with four neighbors based on Kempe transformations.

because then $\mathcal{L}_H$ is normal and by Theorem 1 it is an antimatroid.

Call $\alpha = (x_1 \dots x_m)$. To show $\alpha \in \mathcal{L}_H$, we must show for each $x_i \in \alpha$, $(\exists U \in H_{x_i})[U \subseteq \{x_1, \dots, x_{i-1}\}]$. But since α is a proper shelling of (V, E), each x_i is of degree n or less in the graph $(V, E) \setminus \{x_1, \dots, x_{i-1}\}$. So clearly $\{x_1, \dots, x_{i-1}\}$ is a member of H_{x_i} and we have our U.

$\leftarrow$: If $(V, \mathcal{L}_H)$ is an antimatroid then it contains a sequence $\alpha = (x_1 \dots x_m)$ mentioning all elements of V by Lemma 2. α encodes one possible way to reduce (V, E), since for any graph $(V, E) \setminus \{x_1, \dots, x_{i-1}\}$, $1 \le i \le m$, there is always a vertex with degree n or less that can be removed:

For $i = 1$ clearly x_1 is one of the vertices of degree n or less: $(\exists U \in H_{x_1})[U \subseteq \emptyset]$ implies $\emptyset \in H_{x_1}$ so x_1 already has degree n or less in the graph (V, E). For $i > 1$, we know that once vertices $x_1, \dots x_{i-1}$ have been removed, x_i has degree n or less, since each $x_i \in \alpha$ obeys $(\exists U \in H_{x_i})[U \subseteq \{x_1, \dots, x_{i-1}\}]$, that is, some subset of the vertices $\{x_1, \dots, x_{i-1}\}$ when removed causes x_i to have degree n or less.

■

4 Antimatroids and the Strong Measure of Progress

We would like a more constructive characterization of what it means to have a *strong measure of progress* for planning, which is a metric that tells us which actions to perform to lead us closer to the goal state. (Parmar 2002b) has already shown that the domains with a strong measure of progress are precisely those lacking deadlocks (states from which it is impossible to reach the goal state), since one can only always make strict progress if there are no pitfalls in which to fall.

We first recall our standard formulation of a strong measure of progress. It represented by a predicate defined over a planning domain. The planning domain is described in terms of a standard situation calculus theory T, with finitely many objects $Objects$ in the domain, with the initial situation S_0 and transition function $res(a, s)$. We assume there is only one fluent symbol $\Phi(\overline{x}, s)$, which poses no restrictions since fluents can be "coded" by extra object tuples. For each action symbol a we assume T contains a successor state axiom of the form

$$\begin{aligned}\Phi(\overline{x}, res(a(\overline{y}), s)) \iff \quad & \gamma_\Phi^+(\overline{x}, a(\overline{y}), s) \vee \\ & \Phi(\overline{x}, s) \wedge \neg\gamma_\Phi^-(\overline{x}, a(\overline{y}), s).\end{aligned} \tag{4}$$

γ_Φ^+ and γ_Φ^- are abbreviations for fluent formulas. $\gamma_\Phi^+(\overline{x}, a(\overline{y}), s)$ details the circumstances which are required to make $\Phi(\overline{x}, res(a(\overline{y}), s))$ true, while $\gamma_\Phi^-(\overline{x}, a(\overline{y}), s)$ are those which make it false. T may additionally contain other axioms such as static constraints. $goal(s)$ abbreviates the fluent formulas which are true in the goal state. Planning involves finding a sequence of actions from S_0 such that $goal$ holds at the end.

Definition 4 (Strong Measure of Progress (Parmar 2002b))
*Let $P(x_1, \dots, x_n, s)$ represent a fluent formula with $x_1, \dots, x_n$ object variables and s a situation variable. P is an n-*ary, strong measure of progress *if:*

$$T \models (\forall s)[\neg goal(s) \Longrightarrow (\exists a)[ext(P, s) \subset ext(P, res(a, s))]], \tag{5}$$

where $ext(P, s) = \{\overline{x} \mid T \models P(\overline{x}, s)\}$ and $\subset$ is strict set inclusion.

(5) says that as long as we are not yet in the goal state, we can always find an action which strictly increases the extension of P. To plan, from S_0 we follow the gradient of P upwards, with respect to set inclusion, to eventually reach a goal, in at most $|Objects^n \setminus ext(P, S_0)|$ steps. After this many actions, P will be true of all object tuples in the domain, and then the contrapositive of (5) means that $goal$ will hold.

Intuitively, there seems to be a strong connection between the strong measure of progress P and shelling structures. P essentially cuts up the space in such a way that after enough (or all) objects are "shelled away" into P, we have reached the goal. An even more obvious intuition comes from the fact that antimatroids are a subclass of greedoids, structures synonymous with (greedy) steepest ascent algorithms, which is exactly what P enables.

Here are some initial forays into this idea:

Definition 5 ($\le_P$ ordering)
Let $\overline{x}$ and $\overline{y}$ be n-tuples. Then we define:

$$\overline{x} \le_P \overline{y} \equiv_{def} T \models (\forall s)[P(\overline{y}, s) \implies P(\overline{x}, s)] \tag{6}$$

$\le_P$ formalizes a dependency between $\overline{x}$ and $\overline{y}$; $P(\overline{y}, s)$ cannot be true unless $P(\overline{x}, s)$ is.

The intuition behind $\le_P$ is that it encodes some kind of [goal] ordering on elements of $Objects$, informing us that $\overline{x}$ should be put into P before $\overline{y}$. If we can show that $\le_P$ is a poset, then we can extract an antimatroid that details the ways to put elements into P. Clearly, $\le_P$ is reflexive and transitive, but anti-symmetry requires that:

$$[T \models (\forall s)[P(\overline{x}, s) \iff P(\overline{y}, s)]] \implies \overline{x} = \overline{y}, \tag{7}$$

that is, if $\overline{x}$ and $\overline{y}$ look the same *in all situations and models under* P, then they must be the same tuple. The only two predicates we care about whilst planning is our strong measure of progress P and $goal$. Hence we can assume that (7) holds – finding a plan in a domain where (7) is false is equivalent to finding a plan in a reduced domain lacking mention of $\overline{y}$, since P cannot distinguish between them. This justification should be further studied, however.

The straightforward way to generate our shelling structure is through alternative precedence structures:

$$
\begin{aligned}
H_x = \quad & \{\{y \in Objects \setminus x \mid y \leq_P x\}\} \\
\mathcal{L}_{\leq_P} \quad & = \{(x_1 \dots x_k) \mid (\forall i \leq k)[x_i \in Objects \wedge \\
& \quad (\exists U \in H_{x_i})[U \subseteq \{x_1, \dots x_{i-1}\}]]\} \\
& = \{(x_1 \dots x_k) \mid (\forall i \leq k)(\forall y \in Objects \setminus x_i) \\
& \quad [y \leq_P x_i \implies y \in \{x_1, \dots x_{i-1}\}]\}
\end{aligned} \tag{8}
$$

We would like to show that $\mathcal{L}_{\leq_P}$ mirrors the workings of P – if we keep performing actions which strictly increase P, then the x_is will be absorbed into P in the order given by a string of $\mathcal{L}_{\leq_P}$. We would also like to show that every string of $\mathcal{L}_{\leq_P}$ corresponds to the objects absorbed into P along some action sequence strictly increasing P (and leading to the goal).

First we make some assumptions about our domain T and its associated measure of progress P: we assume $Objects = \{x_1, \dots, x_n\}$ and the goal formula mentions each of the objects: $goal(s) \equiv P(x_1, s) \wedge \dots \wedge P(x_n, s)$. Also assume that $ext(P, S_0)$ is empty, that is, $\neg(\exists x \in Objects)[P(x, S_0)]$. Finally, we assume that S_0 is complete with respect to all fluents; that is for every fluent we know either $F(S_0)$ or $\neg F(S_0)$ holds. Because of the form of our successor state axioms, this guarantees that all facts about every reachable situation is known and that our theory is complete.

Definition 6 ($\mathcal{L}_{P^+traj}$.)
Define the structure $\mathcal{L}_{P^+traj}$ to be the set of strings which record the order in which elements of $Objects$ are added to P along any sequence of actions, starting from S_0, which strictly increase P.

For simplicity, we assume that for each P-increasing action a, *exactly one element of $Objects$* is added to P, courtesy of the following lemma:

Lemma 7 (Simplifying strong measures of progress)
Let T be a domain theory with a strong measure of progress P and objects $Objects$. We can always define another measure of progress Q, on an alternate domain $Objects'$, such that for each P-increasing action a at s we add exactly one element to the extension of Q.

Proof: *The idea is to think of the sets of elements that are added to P's extension as elements in their own right. We visualize the set of elements in $ext(P, res(a, s))$, where $ext(P, res(a, s)) \supset ext(P, s)$, as two elements: $ext(P, s)$, and $ext(P, res(a, s)) - ext(P, s)$. Let $Objects' = 2^{Objects}$. Let $y_1 = ext(P, s)$ and $y_2 = ext(P, res(a, s)) - ext(P, s)$. The following axioms guarantee that $Q(y_1, s) \wedge Q(y_1, res(a, s)) \wedge Q(y_2, res(a, s))$ and for no other $y \in Objects'$.*

$$
\begin{aligned}
Q(y, S_0) \quad & \equiv \quad y = ext(P, S_0) \\
Q(y, res(a, s)) \quad & \equiv \quad Q(y, s) \vee \\
& \quad\quad y = ext(P, res(a, s)) - ext(P, s)
\end{aligned} \tag{9}
$$

■

Continuing, we ask two questions: when does $\mathcal{L}_{\leq_P} \subseteq \mathcal{L}_{P^+traj}$ and $\mathcal{L}_{P^+traj} \subseteq \mathcal{L}_{\leq_P}$? Ultimately we would like to know when $\mathcal{L}_{\leq_P} = \mathcal{L}_{P^+traj}$, that is, when the shelling of the partial order $\leq_P$ encapsulates all the P-increasing plans, or more succinctly, when P generates an antimatroid structure under $\leq_P$. It turns out the first question holds for our assumptions, but not the second:

Theorem 3 ($\mathcal{L}_{P^+traj} \subseteq \mathcal{L}_{\leq_P}$)
The order that elements are added into P starting from S_0 respects the partial order $\leq_P$.

Proof: *Let $\alpha = (x_1 \dots x_k) \in \mathcal{L}_{P^+traj}$ be the sequence of objects which are added to P along some P-increasing path from S_0. Assume otherwise that $\alpha \notin \mathcal{L}_{\leq_P}$, that there are elements $x_i <_\alpha x_j$ but $x_j \leq_P x_i$. Since x_i occurs before x_j, there is a situation s_1 where $P(x_i, s_1) \wedge \neg P(x_j, s_1)$. But $x_j \leq_P x_i \equiv (\forall s)[P(x_i, s) \implies P(x_j, s)]$, a contradiction.* ■

The other direction, that $\mathcal{L}_{\leq_P} \subseteq \mathcal{L}_{P^+traj}$, requires more assumptions. We must show that any shelling of the poset $(Objects, \leq_P)$ is realized by some plan which follows the directives of P. In other words, if $(x_1 \dots x_k) \in \mathcal{L}_{\leq_P}$ then we must show that $P(x_1, res(a, S_0))$ is possible for some a, then $P(x_1, res(a', res(a, S_0))) \wedge P(x_2, res(a', res(a, S_0)))$ for another a', etc. While it is true that eventually all $x_1, \dots, x_n$ will be added to P, we do not know enough about P to guarantee that we can emulate the order given by α. Some possible pitfalls include not having the preconditions be met, or not being able to find an action a to add x_i to P without removing one of $x_1, \dots x_{i-1}$.

It helps to study domains which do have this antimatroid property. The Kitchen Cleaning domain described below has $\mathcal{L}_{P^+traj} = \mathcal{L}_{\leq_P}$. From (Parmar 2002b):

> [C]leaning any object makes it clean. However, cleaning the stove or fridge dirties the floor, and cleaning the fridge generates garbage and messes up the counters. Cleaning either the counters or floor dirties the sink. The goal is for all the appliances to be clean and the garbage emptied.

There is an obvious strong measure of progress P here:

$$
\begin{aligned}
P(fridge, s) & \Longleftrightarrow Clean(fridge, s) \\
P(stove, s) & \Longleftrightarrow Clean(stove, s) \\
P(counters, s) & \Longleftrightarrow Clean(counters, s) \wedge P(fridge, s) \\
P(floor, s) & \Longleftrightarrow Clean(floor, s) \wedge P(fridge, s) \wedge \\
& \quad\quad P(stove, s) \\
P(garbage, s) & \Longleftrightarrow Empty(garbage, s) \wedge P(fridge, s) \\
P(sink, s) & \Longleftrightarrow Clean(sink, s) \wedge P(floor, s) \wedge \\
& \quad\quad P(counters, s),
\end{aligned} \tag{10}
$$

leading to the partial order shown in Figure 3.

Theorem 4 ($\mathcal{L}_{\leq_P} = \mathcal{L}_{P^+traj}$ for Kitchen Cleaning)
For the Kitchen Cleaning domain, any shelling order of $\leq_P$ has an accompanying P-increasing plan with the same element sequence.

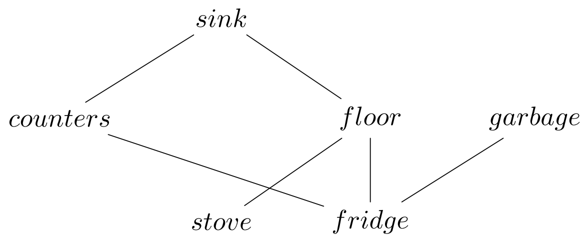


Figure 3: $\leq_P$ for Kitchen Cleaning.

Proof: *Say instead there was a shelling order $\alpha \in \mathcal{L}_{\leq_P}$ and $\alpha \notin \mathcal{L}_{P+traj}$. This means α is a possible shelling of the poset in Figure 3, but no plan exists which would strictly increase P and produce the elements in that order. Identify the first element $x_i \in \alpha$ for which a plan cannot be found. x_i cannot be either $stove$ or $fridge$, since these are always achievable and will not remove elements from P. If x_i were the $floor$ then both $fridge$ and $stove$ must be before x_i in α, in order to respect the order, and therefore in P. In that case, $floor$ is achievable, without threatening to take elements out of P. If $x_i = counters$, then $fridge$ would have to precede it, and then it would be achievable without causing any problems. Similarly, if x_i were $garbage$, $fridge$ would already have to be before x_i, hence put into P already, so that we are free to achieve $garbage$. Finally, if $x_i = sink$, then $counter$, $floor$, $stove$, and $fridge$ would have to be before it, therefore in P, and we are free to clean the $sink$.*

■

It seems that the Kitchen Cleaning example above exhibits the antimatroid property solely because it avoids the two previously mentioned pitfalls that could prevent $\mathcal{L}_{\leq_P} \subseteq \mathcal{L}_{P+traj}$: The first is it does not have to worry about preconditions as any of the actions which achieve part of the $goal$ are immediately achievable; we can put any $x \in Objects$ in P, in any situation. The second has to do with how P orders goals so that they do not get clobbered. Say putting x into P takes some other element w out of P. Then it must be the case that $x \leq_P w$. The intuition is that if achieving x could harm w, then we do it before w, so that we won't have to redo w. For example, with Kitchen Cleaning, cleaning either the $stove$ or $fridge$ will dirty the $floor$, so to prevent this, both $stove$ and $fridge$ are $\leq_P floor$.

Theorem 5 (When $\mathcal{L}_{\leq_P} = \mathcal{L}_{P+traj}$.)
Let T be a domain theory such that $goal(s) \equiv (\forall x) P(x, s)$ and $ext(P, S_0) = \emptyset$. Then if:

1. *accomplishing P for any x is immediately possible, that is:*

$$(\forall x\ s)(\exists a) P(x, res(a, s)),\ and \tag{11}$$

2. *when putting x into P takes w out of P, then $x \leq_P w$:*

$$(\forall s\ a)\ [P(x, res(a, s)) \wedge P(w, s) \wedge \neg P(w, res(a, s)) \implies x \leq_P w], \tag{12}$$

then $\mathcal{L}_{\leq_P} = \mathcal{L}_{P+traj}$.

Proof: *By Theorem 3, it is enough to show that $\mathcal{L}_{\leq_P} \subseteq \mathcal{L}_{P+traj}$. Pick any $\alpha = (x_1 \dots x_n) \in \mathcal{L}_{\leq_P}$ but assume instead that $\alpha \notin \mathcal{L}_{P+traj}$. Let $x_i \in \alpha$ be the first element such that $(x_1 \dots x_{i-1}) \in \mathcal{L}_{P+traj}$ but $(x_1 \dots x_{i-1} x_i) \notin \mathcal{L}_{P+traj}$. This means that we have a set of actions that add $x_1, \dots, x_{i-1}$ to P in that order, up to the situation s_{i-1}, but x_i cannot be a witness for extending P further. Formally:*

$$\begin{aligned} &\neg(\exists a)[(\forall w)[P(w, s_{i-1}) \implies P(w, res(a, s_{i-1}))] \wedge \\ &\quad \neg P(x_i, s_{i-1}) \wedge P(x_i, res(a, s_{i-1}))] \\ \iff &(\forall a)[(\exists w)[P(w, s_{i-1}) \wedge \neg P(w, res(a, s_{i-1}))] \vee \\ &\quad P(x_i, s_{i-1}) \vee \neg P(x_i, res(a, s_{i-1}))] \end{aligned} \tag{13}$$

Also, since $\mathcal{L}_{P+traj}$ records the order of how all *the elements of $Objects$ are added to P, we know that $(x_1 \dots x_{i-1}) \in \mathcal{L}_{P+traj}$ means that they are exactly the elements in P at s_{i-1}: $(\forall y \in Objects)[y \in \{x_1, \dots, x_{i-1}\} \iff P(y, s_{i-1})]$. Now we split on the two cases on whether $x_i = x_1$.*

1. *If $x_i = x_1$, then we have*

$$\begin{aligned} (\forall a)[\quad &(\exists w)[P(w, S_0) \wedge \neg P(w, res(a, S_0))] \vee \\ &P(x_1, S_0) \vee \\ &\neg P(x_1, res(a, S_0))] \end{aligned} \tag{14}$$

   Since $ext(P, S_0) = \emptyset$, both the first two disjuncts of (14) are false. Therefore we infer that $(\forall a) \neg P(x_1, res(a, S_0))$. But this is impossible by assumption (1).

2. *Otherwise, say $x_i \neq x_1$. Pick the action a_{x_i} such that $P(x_i, res(a_{x_i}, s))$. From (13), we know either $(\exists w)[P(w, s_{i-1}) \wedge \neg P(w, res(a_{x_i}, s_{i-1}))]$, or $P(x_i, s_{i-1}) \vee \neg P(x_i, res(a_{x_i}, s_{i-1}))$. The second disjunct is not true; neither is it the case that $P(x_i, s_{i-1})$, as that would require $x_i \in \{x_1, \dots, x_{i-1}\}$ or that α contains a repetition which is impossible, as $\alpha \in \mathcal{L}_{\leq_P}$ which is simple. And of course by definition of a_{x_i} we know $P(x_i, res(a_{x_1}, s_{i-1}))$. It remains to show that the first disjunct is false.*
   If it were true, then $P(w, s_{i-1})$ and given $\neg P(x_i, s_{i-1})$, we know that $\neg x \leq_P w$. But we also know that w is some element in $\{x_1, \dots, x_{i-1}\}$ such that $\neg P(w, res(a_{x_i}, s_{i-1}))$. Given that $P(x_i, res(a_{x_i}, s_{i-1}))$ we can infer from the second assumption that $x \leq_P w$, a contradiction.

■

The gist of Theorem 5 is that $\leq_P$ is too weak and restrictive to characterize the goal dependencies inherent in P. It is not strong enough to characterize how preconditions may depend upon each other, nor how one goal may clobber another. Furthermore it gives little context-specific information, as it is a metric that must hold in *all* situations and models. A more general and perhaps fruitful means of studying P instead is by means of alternative precedence structures. Already this paradigm has a built-in intuition that the sets for a given H_x represent the possible *enablers* for x, that is, if the elements in some $U \in H_x$ have been completed, then one can complete x. However, instead of constructing these H_xs from facts that are true in *all models* and *all situations* (as $\leq_P$ is defined), we instead examine *each* model and *each* situation, and define $U \in H_x \iff$ all the elements in U co-occur with x in some pre-goal situation s and model $\mathfrak{M}$ of T. The advantage of this approach is that we do not require models to be categorical, which means we can drop the requirement that S_0 is complete. Studying this means of representation, and how it correlates with P, is an avenue for future work.

5 More Ideas Using Shelling

Let us represent a planning problem by the tuple $(S_0, A, Objects, G, \gamma_\Phi^+, \gamma_\Phi^-, \alpha)$, where A is the set of actions, and for ease of notation we let $G = \{x_1, \ldots, x_k\} \subseteq Objects$ encode the the goal formula as $goal(s) \equiv \Phi(x_1, s) \wedge \ldots \wedge \Phi(x_k, s)$. The last part of the tuple, α, is a placeholder meant to represent a partial plan inherited from previous transformations. We can think of planning as the gradual shelling over the space of these tuples, using various transformations. In this section we discuss two of them:

1. We say a goal $\Phi(x_i, s)$ is *init-minimal* if for some action $a_0 \in A$,

$$\gamma_\Phi^+(x_i, a_0, S_0) \wedge (\forall a, s)\neg\gamma_\Phi^-(x_i, a, s) \tag{15}$$

   that is, we can make $\Phi(x_i, \cdot)$ true in $res(a_0, S_0)$ and know that no matter what happens, this part of the goal won't ever become false (get clobbered).[2] With such an x_i we can reduce our planning problem to $(res(a_0, S_0), A, Objects, G \setminus x_i, \gamma_\Phi^+, \gamma_\Phi^-, a_0\alpha)$.

2. A goal $\Phi(x_j, s)$ is *goal-maximal* if

$$(\forall s)(\exists a_j)\gamma_\Phi^+(x_j, a_j, s) \wedge (\forall y \in G \setminus x_j)[\gamma_\Phi^+(y, a_j, s) \vee \neg\gamma_\Phi^-(y, a_j, s)]. \tag{16}$$

   This is much like the map-coloring problem in that no matter in what situation we end up, we can always find an action a_j to accomplish x_j without violating any already-achieved goals. If we can find a goal-maximal goal $\Phi(x_j, s)$, then our plan reduces to $(S_0, A, G \setminus x_j, \gamma_\Phi^+, \gamma_\Phi^-, \alpha a_j)$.

These are only two possible ways to shell this space. There are many other transformations that are possible, and it is most likely that the space will not be a proper antimatroid, as not all elements will be shelled (but perhaps a matroid). But even if the transformations do not lead to a plan without search, they could at least lead to a difficult subcomponent minimal in size.

[2] Really, we only have to worry that it does not get false over any possible solution path. By expanding the restriction to the entire situation tree we make our requirement harder to satisfy but much easier to handle in logic.

6 Sequences of Actions and the Strong Measures of Progress

One simple adaptation to the strong measure of progress is to quantify over *sequences* of actions:

$$T \models (\forall s)[\neg goal(s) \Longrightarrow (\exists \overline{a})[ext(P, s) \subset ext(P, res(\overline{a}, s))]] \tag{17}$$

This would enable us to quantify progress on a much larger scale. Instead of having to logically navigate every possible plateau and local minimum on our path to the goal, we could simply use sequences of actions, or macro-operators, to "jump" over them. The whole purpose of having a strong measure of progress is to describe in broad terms how to get directly to the goal – having sequences of actions as objects accomplishes this directly. If properly formulated, we might even be able to have measures of progress for domains with deadlocks, if we can assure that all sequences purposely avoid deadlocks.

However, we must formulate equations like (17) very carefully. If $\overline{a}$ ranges over all sequences of actions, then the equation above is a tautology – simply take $\overline{a}$ to be any viable plan from s. We must carefully decide which action sequences to invite, as they can easily break the constructive nature of the measure of progress and ruin our party.

In fact, as soon as we add a sequential operator that concatenates actions to our domain, we immediately break (17), as that allows us to formulate the true plan to the goal. Golog (Levesque *et al.* 1997), without the concatenation operator ";", may serve as a good language for talking about macro-operators, especially because it has the ability to sense fluents which allows us to distinguish between (and compactly represent) different conditions. This topic of research is something that must be further investigated.

The Towers of Hanoi is a domain which can benefit from a proper set of macro-operators. In this problem, there are three pegs: A, B, and C, and n disks, with disk d_i of size i. Initially, the disks are stacked in order of increasing size on peg A, largest disk on the bottom. The goal is to find a plan which moves the entire stack to peg C, given that one can move only the topmost disk d_t of a peg, and that it can only be placed either on top of a disk larger than itself, or an empty peg.

There are many solutions to this problem, outlined in (Syropoulos 2002), which include complicated bit encodings.[3] We present the traditional recursive encoding for arbitrary n:

[3] These encodings are strong measures of progress!

$$p = solve(n, A, B, C) \iff (n = 0 \land p = \emptyset) \lor (n \neq 0 \land p = solve(n-1, A, C, B); move(A, C); solve(n-1, B, A, C)) \tag{18}$$

The problem of moving n disks from A to C is reduced to moving the first $n-1$ to B, then moving the largest disk from A to C, and then moving the $n-1$ to C. The recursive solution to Towers of Hanoi involves two new ideas; first, a macro-operator $solve(n, X, Y, Z)$ which has an inductive parameter, and careful (as in not arbitrary) sequential combination. The macro-operator abbreviates a set of moves we would rather not enumerate, by induction on its numeric parameter.

If S is the set of actions $\{solve(n-1, A, C, B), move(A, C), solve(n-1, B, A, C)\}$, then quantification of $\overline{a}$ over S in (17) would lead to a simple formulation of P:

$$\begin{aligned}
P(0,s) \iff & [On(d_n, A, s) \land \\
& On(d_1, d_2, s) \land \ldots \land On(d_{n-2}, d_{n-1}, s) \land \\
& On(d_{n-1}, B, s)] \lor P(1, s) \\
P(1,s) \iff & [On(d_n, C, s) \land \\
& On(d_1, d_2, s) \land \ldots \land On(d_{n-2}, d_{n-1}, s) \land \\
& On(d_{n-1}, B, s)] \lor P(2, s) \\
P(2,s) \iff & [On(d_n, C, s) \land \\
& On(d_1, d_2, s) \land \ldots \land On(d_{n-2}, d_{n-1}, s) \land \\
& On(d_{n-1}, C, s)],
\end{aligned} \tag{19}$$

It would be interesting to see what better measures of progress are obtained if we added similar kinds of macro-operators to our other planning domains. One immediate, curious, side-effect of adding such macros is that they avoid an explosion in the description of plans. For Towers of Hanoi, a plan for moving n disks has length $\Theta(2^n)$, and since the m-ary strong measure of progress can encode plans of at most length n^m, m must be $O(n \log_n 2)$, a function of n! We contrast this with the strong measures of progress developed in (Parmar 2002b) for Blocksworld and Logistics World, which has a fixed arity regardless of the number of objects in the domain. But if we add macros to the Towers of Hanoi, we can get back a fixed arity measure of progress, in fact, one whose definition does not change over problem instances, as shown in (19). One concept we have glossed over however, is how to efficiently utilize the more concise measure of progress in (19) and in conjunction with the definitions of macro-operators in (18).

7 Conclusions and Discussion

Finding out when map coloring can be done without backtracking is an illustrative example for our motivations for planning – we want to identify the obvious next action to perform by reasoning, not searching. The strong measure of progress is supposed to encode this, but how can we discover these measures in practice? Antimatroids provide some hint as to the structure which is involved, and we have not yet exhausted the subject.

The reduction we performed to do backtrack-free map coloring is applicable to CSPs, and it would be interesting to see how concepts of arc- and node-consistency relate to antimatroids. Since CSPs are difficult, we do not imagine that antimatroids will entirely solve every CSPs, but at least they can identify and quickly solve some isolated regions of the problem. We also want to apply this idea of identifying and solving the easy parts of a problem to planning.

Section 5 makes some forays into how we can shell subgoals either after the initial state or before the goal state. An intriguing notion not explored here would be how one can add additional, supporting subgoals, much as is done in partial-order planning, in order to aid the shelling process. (We may need to add elements to our structures to make them look like antimatroids.) Adding macro-operators to aid in the shelling might also be useful.

We find the idea of adding some carefully constructed set of macro-operators to simplify our measures of progress to be very intriguing. This is the tack that is used in (Korf 1985), where our measure of progress corresponds to some total ordering of the conjuncts of a goal. The inductive feel of the Towers of Hanoi problem could be generalized and used to make macros for other planning problems. Using Golog programs as macros sounds especially interesting, because they can sense and react accordingly, making macros even more expressive.

Some topics we would like to explore in the future include how to expand these results to domains that include deadlocks. Also, there appears to be a duality between having to enumerate subgoals, and adding macro-operators in planning. For example, in Towers of Hanoi, without macros our strong measure of progress's arity grows in the number of disks, whereas adding the macro $solve(n, X, Y, Z)$ ensures it remains of constant size.

A final issue that not discussed here is how we can *reformulate planning problems* to make them more amenable to solve. This could involve something as low-level as rewriting the theory so that the $\gamma^{\pm}_{\Phi}(\overline{x}, a, s)$s do not depend on s, or some overall transformation on the fluents of the domain so that certain dependencies are more perspicuous (think of finding a basis in linear algebra).[4] A great result would be a formal connection between antimatroids and means-ends analysis.

8 Acknowledgments

We thank the anonymous reviewers for their helpful comments. This research has been partly supported by SNWSC contract N66001-00-C-8018.

References

Appel, K., and Haken, W. 1977a. Every planar map is four colorable. part i. discharging. *Illinois Journal of Mathematics* 21:429–490.

[4]Gaussian elimination, one of the methods for finding a basis, gives rise to a greedoid.

Appel, K., and Haken, W. 1977b. Every planar map is four colorable. part ii. reducibility. *Illinois Journal of Mathematics* 21:491–567.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artificial Intelligence* 69(1-2):165–204.

Fox, M., and Long, D. 2001. Hybrid STAN: Identifying and managing combinatorial optimisation sub-problems in planning[5]. In *IJCAI-01*.

Kempe, A. B. 1879. On the geographical problem of the four colours. *American Journal of Mathematics* 2(3):193–200.

Korf, R. 1985. *Learning to Solve Problems by Searching for Macro Operators.* Ph.D. Dissertation, Carnegie Mellon University.

Korte, B., and Lovász, L. 1981. Mathematical structures underlying greedy algorithms. In *Fundamentals of Computation Theory*, volume 117 of *Lecture Notes in Computer Science*, 205–209. Springer-Verlag.

Korte, B.; Lovász, L.; and Schrader, R. 1991. *Greedoids*. Springer-Verlag.

Levesque, H. J.; Reiter, R.; Lesprance, I.; Lin, F.; and Scherl, R. B. 1997. Golog: A logic programming language for dynamic domains. *Journal of Logic Programming* 31(1–3):59–83.

Long, D., and Fox, M. 2000. Automatic synthesis and use of generic types in planning. In *AIPS-00*, 196–205.

McCarthy, J. 1982. Coloring maps and the Kowalski doctrine[6]. Technical Report STAN-CS-82-903, Dept Computer Science, Stanford University. AIM-346.

Parmar, A. 2002a. A Logical Measure of Progress for Planning (Technical Report)[7]. Technical report, FRG.

Parmar, A. 2002b. A Logical Measure of Progress for Planning[8]. In *AAAI/IAAI*, 498–505. AAAI Press. Technical report is (Parmar 2002a).

Robertson, N.; Sanders, D. P.; Seymour, P.; and Thomas, R. 1995. The Four Color Theorem[9]. Webpage.

Syropoulos, A. 2002. The Towers of Hanoi[10]. Webpage.

[5] http://www.dur.ac.uk/˜ dcs0www/research/stanstuff/Papers/sigpaper.ps

[6] http://www-formal.stanford.edu/jmc/coloring.html

[7] http://www-formal.Stanford.edu/aarati/techreports/aaai-2002-t r.ps

[8] http://www-formal.Stanford.edu/aarati/papers/aaai-2002.ps

[9] http://www.math.gatech.edu/˜thomas/FC/fourcolor.html

[10] http://obelix.ee.duth.gr/˜apostolo/TowersOfHanoi/