On the utility of Plan-space (Causal) Encodings

Amol D. Mali & Subbarao Kambhampati*
Dept. of Computer Science & Engineering
Arizona State University, Tempe, AZ 85287-5406
Email: {amol.mali, rao}@asu.edu; URL: rakaposhi.eas.asu.edu/yochan.html

Abstract

Recently, casting planning as propositional satisfiability has been shown to be a very promising technique for plan synthesis. Although encodings based both on state-space planning and on plan-space (causal) planning have been proposed, most implementations and trade-off evaluations primarily use state-based encodings. This is surprising given both the prominence of plan-space planners in traditional planning, as well as the recent claim that lifted versions of causal encodings provide the smallest encodings. In this paper we attempt a systematic analytical and empirical comparison of plan-space (causal) encodings and state-space encodings. We start by pointing out the connection between the different ways of proving the correctness of a plan, and the spectrum of possible SAT encodings. We then characterize the dimensions along which causal proofs, and consequently, plan-space encodings, can vary. We provide two encodings that are much smaller than those previously proposed. We then show that the smallest causal encodings cannot be smaller in size than the smallest state-based encodings. We shall show that the "lifting" transformation does not affect this relation. Finally, we will present some empirical results that demonstrate that the relative encoding sizes are indeed correlated with the hardness of solving them. We end with a discussion on when the primacy of traditional plan-space planners over state-space planners might carry over to their respective SAT encodings.

1 Introduction

Impressive results have been obtained by casting planning problems as propositional satisfiability [Kautz & Selman 96]. The general idea of this paradigm is to construct a disjunctive structure of size k that contains all possible action sequences of length k that can potentially solve the problem. The problem of checking if there exists a sequence that actually solves the problem is posed as an instance of satisfiability checking. The encoding contains constraints that must hold for any specific sequence to be a solution. Informally, the constraints specify lines of proof that must hold for a sequence to be a solution to the given planning problem. In classical planning, there are two general ways of "proving" that a sequence of actions solves a planning problem $[I, G]$: (i) The "state space" methods that essentially try to progress the initial state I (or regress the goal state G) through the sequence to see if the goal state (or initial state) is reached. (2) The "plan space" or "causal" methods that attempt to check if every goal and precondition of every action is effectively established (i.e., there exists some preceding action that contributes that condition, and the condition survives up to the needed step).

Although encodings based on both state space proofs and plan space proofs have been considered in the literature [Kautz *et. al.* 96], most implementations and trade-off studies have concentrated almost exclusively on the state-based encodings [Ernst *et. al.* 97; Kautz & Selman 96]. This is indeed surprising given that the only published theoretical study of causal encodings [Kautz *et. al.* 96] is quite supportive of the relative utility of causal encodings. That study claimed that the lifted version of causal encoding is asymptotically the smallest of all encodings including state-based encodings.

In this paper, we report on a theoretical and empirical study of the utility of causal (plan space) encodings. We make the following contributions:

- We show that there are many variations of plan space encodings that, roughly speaking, differ in the specific ways they carry out the causal proofs over action sequences. These variations are interesting as they can have significant impact on the size of the encoding.
- We analyze the sizes of our best causal encodings, and show that they have significantly better asymptotic size characteristics than the only causal encoding that has been previously described in the literature [Kautz *et. al.* 96].
- We compare the sizes of our causal encodings with the sizes of the best state-based encodings from the literature, and note that causal encodings are in fact never strictly smaller than best state-based encodings.
- We provide a theoretical argument as to why no type of causal encoding can be smaller than the best state-based encoding.
- We show that the "lifting" transformation *does not* change this dominance of causal encodings by the state-based encodings.
- We describe results of empirical studies that show that the hardness of solving the encodings is in fact correlated with the encoding sizes. Specifically, our studies

*Copyright(c) 1999, American Association for Artificial Intelligence (www.aaai.org). All rights reserved. This research is supported in part by NSF young investigator award (NYI) IRI-9457634, NSF grant IRI-9801676. ARPA/Rome Laboratory planning initiative grant F30602-95-C-0247, and AFOSR grant F20602-98-1-0182.

show both that our best causal encodings are better than the causal encoding previously presented in the literature, and that even our encodings are dominated by the best state-based encodings.

- We put our results in perspective by considering the reasons why plan space (or causal) approaches were found to be superior in traditional planning, and explaining why those reasons do not hold in the planning as satisfiability framework. We will also show variations to the planning as satisfiability framework where causal encodings have utility.

The paper is organized as follows. In section 2, we explain our notation used for representing the planning constraints and explain some key constraints from the state-based encoding. In section 3, we report several variants of the causal encoding of [Kautz *et. al.* 96] and show that some of our variants are smaller. We establish limitations on the reduction in the size of the causal encodings in section 4. In section 5, we show that the lifting transformation does not change the relationship between the sizes of the state-based and the causal encodings. Section 6 presents the results of our empirical studies on various encodings. Section 7 puts our results in perspective, and Section 8 presents the conclusions.

2 Background

As we mentioned earlier, compiling planning into satisfiability checking involves constructing a disjunctive structure of k steps, and writing down the set of constraints that must hold for any action sequence belonging to this structure to be a valid plan for the given problem. The encoding is thus specified in such a way that it has a model if and only if there exists a provably correct plan of k steps. If no model is found, it means that any plan for the problem must be longer than k steps. Accordingly, a new encoding is generated by increasing the value of k. We shall start by describing some common notation, and then go on to describe the basic ideas of the state-based encodings.

2.1 Notation

p_i denotes a step and o_j denotes a ground action. $o_j(t)$ denotes that the action o_j occurs at time t. k is the number of plan steps and U is the set of pre-condition and effect propositions u_j in the domain. $u_j(t)$ denotes that the proposition u_j is true at time t. O is the set of non-null ground actions in the domain. $(p_i = o_j)$ denotes the step→action mapping. ϕ denotes the null action (no-op) that does not cause any change to the world state. a_i denotes a fluent from the goal state (partially described) G. G is assumed to be $(a_1 \wedge a_2 \wedge a_3 ... \wedge a_h)$. F denotes the goal state step (goal state can be viewed as a step with preconditions same as the goals and no effects). I denotes the completely specified initial state and l_i denotes a fluent true in the initial state. $\mid I \mid$ denotes the number of fluents true in I. $p_i \stackrel{f}{\rightarrow} p_j$ denotes a causal link where p_i adds (makes true) the condition f, p_j needs it and p_i precedes p_j. A_j, R_j, D_j denote the number of add effects, pre-conditions and delete effects of the action o_j respectively. a_{js}, r_{jt}, d_{jq} denote the individual add effects, pre-conditions and delete effects of the action o_j respectively. $Adds(p_i, u_j), Needs(p_j, u_j)$ and $Dels(p_q, u_j)$ respectively denote that the steps p_i, p_j, p_q add, need and delete u_j. $p_i \prec p_j$ denotes that the step p_i precedes the step p_j. Note that we distinguish total order on steps from contiguous order, e.g. the steps p_1, p_2, p_3, $p_1 \prec p_2, p_2 \prec p_3$ are totally ordered, but a new step p_4 can occur between them, e.g. $p_1 \prec p_4, p_4 \prec p_2$. If the steps are contiguous, no new steps can be inserted between them (although, as we shall see in Section 7, this distinction is immaterial for from-scratch planning). We denote an encoding of a planning problem P, by $E_i(P)$.

We define an encoding $E_i(P)$ to be strictly larger than an encoding $E_j(P)$ if and only if either $E_i(P)$ has higher number of variables or clauses or literals (sum of the lengths of the clauses) than $E_j(P)$, with other parameters (#variables, #clauses and #literals) being at least as high or higher.

2.2 Basics of a State-based Encoding

State-based encodings are based on the ideas of proving the correctness of a plan using progression or regression. The latter involves simulating the regression of the goal state over the last step of the plan, and regressing the resulting state over the last but one step etc. Correctness of the plan holds as long as the final state resulting from this process is subsumed by the initial state. An important notion in the state-based encodings is thus the availability of the world state at each time step. The clauses in a state-based encoding capture the following constraints: Any of the $\mid O \mid$ actions from the domain may occur at any of the k time steps from the interval $[0, k-1]$ and an action that occurs at time t implies the truth of its pre-conditions at t and the truth of its effects at $(t+1)$. The initial state is true at time 0 and the goal must be true at time k. Conflicting actions (one action deleting the pre-condition or effect of another or needing negation of pre-condition of another) cannot occur at the same time step. In addition we need frame axioms that capture the persistence of fluents. This can be done by the "classical frame" axioms that state that a fluent u_j remains unchanged in the interval $[t, t+1]$ if the action occuring at t doesn't have u_j in its add or delete list. A more efficient alternative is to use "explanatory frame axioms" which state that if the truth of a fluent u_j changes over an interval $[t, t+1]$, some action changing that truth must occur at t. We restrict our attention to the state-based encoding with explanatory frame axioms, as this encoding has been shown to have lower size, as well as faster solvability [Ernst *et. al.* 97]. Because of the representation of all step-action bindings, the use of explanatory frame axioms, and the fact that an action implies the truth of its pre-conditions and effects, the state-based encoding contains $O(k*(\mid O \mid + \mid U \mid))$ clauses, $O(k*(\mid O \mid + \mid U \mid))$ variables and a total of $O(k* \mid O \mid * \mid U \mid)$ literals.

3 Causal (Plan-space) Encodings

The plan space (causal) encodings are based on the ideas of proving the correctness of a plan using causal reasoning about the establishment and preservation of goals and the preconditions of individual actions. The correctness of the plan is proved by ensuring that (i) every precondition r of every step s is made true by some step s' that precedes s (establishment) and (ii) r remains true, when it is needed immediately before s (declobbering). There are several variants of "causal proof," based on how the two conditions above are guaranteed. The popular approach for establishment involves associating a "causal link" $s' \stackrel{r}{\rightarrow} s$, with every precondition r of s [McAllester & Rosenblitt, 91]. A problem with this approach, as we shall see below, is that encodings based on it will have a quadratic number of variables corresponding to causal links. An alternative is to dispense with causal links, and post constraints to ensure that for each precondition that there is a contributor.

To ensure that the established condition is available at the needed step, we might either require that it not be deleted by any possibly intervening step ("interval protection"), or that for every deleting step, there be a re-establishing step ("white-knight protection") [Kambhampati *et. al.*, 95]. Finally, the specific implementations of establishment and declobbering conditions depend on the ordering between the steps in the plan. Traditionally, plan-space proofs were associated with the so-called "partial-order" planners [McAllester & Rosenblitt, 91], where the steps in the plan are partially ordered. Such an ordering was important since those planners incrementally introduced steps anywhere in the plan. As we shall see below, partial ordering is expensive to encode because of the need for encoding transitive ordering relations between the steps. This, coupled with the fact that in setting up SAT encodings, we are not interested in "inserting" new steps into an existing plan, suggests that we pursue more restrictive ordering schemes, including contiguity ordering (where the relative positions of each of the steps in the encoding are fixed *a priori*). Since all possible step→action mappings are represented in any encoding, the models of an encoding with contiguous steps are exactly same as the models of an encoding with partially ordered steps.

Given the choice in the way establishment and declobbering are realized, and the specific ordering scheme used in the encoding, we have a spectrum of possible encodings. Only one of these encodings, corresponding to casual link based establishment, interval protection based declobbering, and partially ordered steps, has been studied previously [Kautz *et. al.*, 96]. We have studied the rest of the variations, and found that several of them have better asymptotic sizes than that in [Kautz *et. al.*, 96]. In the following, we will present and analyze the variation corresponding to that studied in [Kautz *et. al.*, 96], as well as two other superior variations in our spectrum.

Before we proceed however, we shall briefly describe the set of axiom schemas that are common to all the variations of causal encodings. Figure 1 lists these schemas formally. Briefly, the first two axiom schemas state that each step in the encoding must be mapped to a single domain action or a no-op. The third schema says that the facts true or false in the initial state are considered to be the effects of step I and the facts specified in the goal state are considered to be the preconditions of step F. The fourth schema says that if a step is mapped to an action, then that step inherits the preconditions and effects of that action. The fifth schema states that the only way a step can add, delete or require a condition is if the condition is added, deleted or required (respectively) by the action that the step is mapped to.

3.1 Causal links, Interval protection & Partial ordering

The first encoding we consider uses causal links for establishment, interval protection for declobbering and assumes that the steps are partially ordered. This variation corresponds to that studied in [Kautz *et. al.*, 96]. The additional axioms (over and above the common ones already shown in Figure 1) that are needed for this encoding are shown in Figure 2. Axiom schema 6 states that each precondition must have a causal

6. $\wedge_{i=1}^{k} \wedge_{j=1}^{|U|} (Needs(p_i, u_j) \Rightarrow (\vee_{q=1, q \neq i}^{k} (p_q \xrightarrow{u_j} p_i) \vee (I \xLeftrightarrow{u_j} p_i)))$

7. $\wedge_{i=1}^{k} \wedge_{j=1, i \neq j}^{k} \wedge_{q=1}^{|U|} (p_i \xLeftrightarrow{u_q} p_j \Rightarrow (Adds(p_i, u_q) \wedge Needs(p_j, u_q) \wedge (p_i \prec p_j)))$

8. $\wedge_{i=1}^{k} \wedge_{j=1, i \neq j}^{k} \wedge_{s=1, s \neq i, s \neq j}^{k} \wedge_{q=1}^{|U|} ((p_i \xLeftrightarrow{u_q} p_j \wedge Dels(p_s, u_q)) \Rightarrow ((p_s \prec p_i) \vee (p_j \prec p_s)))$

9. $\wedge_{i=1}^{k} \wedge_{j=1, i \neq j}^{k} \wedge_{s=1, s \neq i, s \neq j}^{k} (((p_i \prec p_j) \wedge (p_j \prec p_s)) \Rightarrow (p_i \prec p_s))$

$\wedge_{i=1}^{k} \wedge_{j=1}^{k} \neg((p_i \prec p_j) \wedge (p_j \prec p_i)), \wedge_{i=1}^{k} \neg(p_i \prec p_i)$

Figure 2: Schemas for the encoding in [Kautz *et. al.*, 96]

link supporting it (with the role of contributor step played by one of the steps in the encoding). The schemas 7 and 8 ensure that the contributor step of a causal link precedes the consumer step, and that if a step is mapped to an action that deletes the condition supported by the causal link, then that step either precedes the contributor or succeeds the consumer. Finally, we also need to add a set of constraints capturing the irreflexiveness, asymmetry and the transitivity of the precedence relation (schema 9).

Since there are $O(k^2 * \mid U \mid)$ causal links each of which may be threatened by $O(k)$ steps, there are $O(k^2 * \mid U \mid)$ variables and $O(k^3 * \mid U \mid)$ clauses in the causal encoding of [Kautz *et. al.* 96] (for threat resolution).

3.2 Causal Links, Interval Protection & Contiguous steps

We now consider the variant that uses causal links for establishment, interval protection for declobbering but assumes that the steps are contiguous. Figure 3 shows the distinguishing schemas of this variant. Since the ordering is contiguous, we can represent it by numbering steps in the encoding successively $1 \cdots k$. There is no need to represent precedence relations, or describe their properties (including the costly transitivity relation). Schema 6 states the requirement that each precondition of each step is supported by a step whose position is before that of the consumer step. The interval protection of causal links (Schema 7) involves ensuring that no step in the positions between those of the contributor and consumer steps is mapped to an action that deletes the supported condition.

For resolving the threats to causal links $p_{i_1} \xrightarrow{f} p_{i_2}$, we need $\frac{k*(k+4)*(k-1)}{6} * \mid U \mid$ clauses. Since the encoding in [Kautz *et. al.*, 96] uses partial ordering instead of contiguity ordering, it needs $k*(k-1)*(k-2)* \mid U \mid$ threat resolving clauses. Although both are asymptotically of the same order ($O(k^3)$), the contiguity relation allows us to achieve a percentage reduction in the number of clauses of $[1 - \frac{(k+4)}{6*(k-2)}] * 100$. As $k \rightarrow \infty$, this reduction tends to 83.33%, which is quite significant.

3.3 No Causal Links, White-knight protection & Contiguous Steps

We now consider a further departure from the encoding in [Kautz *et. al.*, 96] by dispensing with causal links for estab-

$$\mathbf{1}.\ \wedge_{i=1}^{k}\ (\vee_{j=1}^{|O|}(p_i = o_j) \vee (p_i = \phi))$$

$$\mathbf{2}.\ \wedge_{i=1}^{k}\ \wedge_{j_1=1}^{|O|}\ \wedge_{j_2=1, j_2 \neq j_1}^{|O|}\ \neg((p_i = o_{j_1}) \wedge (p_i = o_{j_2}))$$

$$\wedge_{i=1}^{k}\ \wedge_{j=1}^{|O|}\ \neg((p_i = o_j) \wedge (p_i = \phi))$$

$$\mathbf{3}.\ \wedge_{s=1}^{|I|}\ Adds(I, l_s), \wedge_{s=1}^{|U-I|}\neg Adds(I, a_{js}), \wedge_{i=1}^{h} Needs(F, a_i)$$

$$\mathbf{4}.\ \wedge_{i=1}^{k}\ \wedge_{j=1}^{|O|}((p_i = o_j) \Rightarrow ((\wedge_{s=1}^{A_j} Adds(p_i, a_{js})) \wedge (\wedge_{t=1}^{R_j} Needs(p_i, r_{jt}))) \wedge (\wedge_{q=1}^{D_j} Dels(p_i, d_{jq})))$$

$$\mathbf{5}.\ \wedge_{i=1}^{k}\ \wedge_{j=1}^{|U|}(Adds(p_i, u_j) \Rightarrow (\vee_{q=1}^{x}(p_i = o_{m_q}))), Adds(o_{m_q}, u_j)$$

$$\wedge_{i=1}^{k}\ \wedge_{j=1}^{|U|}\ (Dels(p_i, u_j) \Rightarrow (\vee_{q=1}^{x}(p_i = o_{m_q}))), Dels(o_{m_q}, u_j)$$

$$\wedge_{i=1}^{k}\ \wedge_{j=1}^{|U|}\ (Needs(p_i, u_j) \Rightarrow (\vee_{q=1}^{x}(p_i = o_{m_q}))), Needs(o_{m_q}, u_j)$$

Figure 1: The schemas common to all causal encodings.

6.

$$\wedge_{i=1}^{k}\wedge_{j=1}^{|U|}(Needs(p_i, u_j) \Rightarrow (\vee_{q=1}^{i-1}(p_q \overset{u_j}{\rightarrow} p_i) \vee (I \overset{u_j}{\rightarrow} p_i)))$$

7.

$$\wedge_{i_1=1}^{k-1}\ \wedge_{i_2=i_1+1}^{k}\ \wedge_{j=1}^{|U|}((p_{i_1} \overset{u_j}{\rightarrow} p_{i_2}) \Rightarrow ((Needs(p_{i_2}, u_j)$$

$$\wedge Adds(p_{i_1}, u_j) \wedge (\wedge_{q=i_1+1}^{i_2-1}\neg Dels(p_q, u_j)))))$$

Figure 3: Schemas for the encoding based on causal link protection with contiguous steps

6.

$$\wedge_{i=2}^{k}\ \wedge_{j=1}^{|U|}\ (Needs(p_i, u_j) \Rightarrow (\vee_{q=1}^{i-1} Adds(p_q, u_j) \vee Adds(I, u_j)))$$

7.

$$\wedge_{i=3}^{k}\ \wedge_{j=1}^{i-2}\ \wedge_{m=1}^{|U|}((Needs(p_i, u_m) \wedge Dels(p_j, u_m)) \Rightarrow (\vee_{q=j+1}^{i-1} Adds(p_q, u_m)))$$

$$\wedge_{j=1}^{k-1}\ \wedge_{m=1}^{h}\ ((Needs(F, a_m) \wedge Dels(p_j, a_m)) \Rightarrow (\vee_{q=j+1}^{k} Adds(p_q, a_m)))$$

Figure 4: Schemas for the causal-link less encoding based white-knight protection and contiguity ordering

lishment, and using white-knight protection for declobbering. We will continue to assume that the steps in the encoding are contiguous (as in Section 3.2). This variant turns out to be the smallest (has the fewest number of clauses, variables and literals) of all the causal encodings in the spectrum of encodings we have considered.

The key schemas of this encoding are shown in Figure 4. The establishment schema 6 eliminates the causal links by only requiring only that each pre-condition of each step p_i must be added by some step whose position precedes p_i. The declobbering schema 7 says that any deleted pre-condition must be re-established. Notice that there is no reference to any particular causal link intervals. Since we are considering steps in a contiguous ordering, this schema generates only $O(k^2 * \mid U \mid)$ clauses, as opposed to $O(k^3 * \mid U \mid)$ in the previous two encodings. Traditional planners that use white-knight protection strategy, such as TWEAK [Chapman, 87] have been found to be inferior to the causal link-based planners because they may establish a condition multiple times [Minton *et. al.* 91]. It is thus interesting to note that the combination of white-knight protection, causal-link-less establishment and contiguous ordering leads to a very compact causal encoding (see also Section 7)!

4 Comparison with State-based Encodings

As mentioned earlier, state-based encodings with explanatory frame axioms have been shown to be smallest among state-based encodings. A comparison of the smallest variant (from section 3.3) of the causal encodings with the state-based encoding with explanatory frame axioms shows that the asymptotic number of variables in both encodings are the same ($O(k * (\mid O \mid + \mid U \mid))$). However, the state-based encoding with explanatory frame axioms has fewer (that is $O(k * (\mid U \mid + \mid O \mid))$) clauses. Hence the state-based encoding with explanatory frame axioms remains smaller than the smallest causal encoding. Indeed, we can view the white knight strategy as an *inefficient version of the explanatory frame axioms*. The regular explanatory frame axioms explain the change of truth of a world state fluent over just the unit time intervals $[t, t+1]$ (the number of these time intervals is $O(k)$), however the white-knight strategy can be seen as explaining this change over all time intervals (the number of these time intervals is $O(k^2)$).

One natural question is whether the dominance of state-based encodings holds irrespective of the specific variant of causal encodings considered. As the result below shows, the relative dominance holds irrespective of the variant of the causal encodings used. The proof is based on the observation that the causal encodings have to consider the truth of conditions over many more time intervals than state-based encodings do.

The important property of a causal proof is its ability to consider the truth of each precondition in isolation from other preconditions. This is achieved by considering all possible establishing actions and all possible ways of protecting (declobbering) those establishments. Since the precondition of an action occurring at time t could have been made true at any time $j \in [0, t]$ any causal encoding will have to refer to a quadratic number of time intervals, their lengths varying from 1 to $(k+1)$ and resolve threats posed by steps occurring in these longer time intervals. This holds irrespective of whether the ordering between actions is partial, total or contiguous.

In contrast, in the state-based encodings, the world state at t serves as the contributor for every pre-condition of every action that occurs at t. Hence a state-based encoding need to refer to only a linear number of time intervals ($(k+1)$ for a k step plan), each of length 1.

The foregoing shows that a causal encoding will always have more clauses than a state-based encoding. It is possible to show that this dominance holds also for the number of variables and the number of literals (sum of clause lengths). Hence we have the theorem:

Theorem 1. *Causal encodings are strictly larger than the smallest state-based encoding.*

5 The effect of "lifting"

[Kautz *et. al.* 96] have argued that the smallest encoding is the "lifted" version of their causal encoding. Lifting is motivated by the fact that number of ground actions is generally combinatorially large. Lifted encodings use only the uninstantiated action schemas and leave it to the solver to decide the instantiations of arguments of the actions, by stating that each argument can be mapped to any of the elements from its domain and some other constraints. The idea is to replace the complexity of solving a larger ground encoding with the complexity of solving a smaller lifted encoding and doing unifications using the ground initial and goal state. To our knowledge, this speculation is not yet validated due to the lack of effective lifted solvers. Nevertheless, in this section, we argue that any potential size improvements from lifting will also apply to the state-based encodings. Specifically, lifted state-based encodings can be proved to be smaller than the lifted causal encodings, as shown next.

In Figure 5, we show the schemas that are required to generate a lifted version of ground state-based planning. The set of lifted actions is denoted by O'. A lifted action is denoted by o'_i, and its lifted add, delete and precondition fluents are denoted by $a'_{ij}, d'_{ij}, n'_{ij}$. U' is the set of lifted pre-conditions and effects and u'_j denotes a lifted fluent from U'. The initial and goal states are ground. Schema 3 that states the explanatory frame axioms, says that if the truth of a proposition changes, some lifted action whose ground version can cause the change must have occurred. Schema 6 states that each action argument variable x_i can take any value c_{ij} from its domain Dom_i and V denotes the set of these arguments.

It can be seen that even the lifted version of the state-based encoding with explanatory frame axioms is smaller than the lifted version of the smallest causal encoding, because the lifted state-based encoding will have $O(k * (\mid O' \mid + \mid U' \mid))$ variables and clauses, but the smallest lifted causal encoding will have $O(k^2 * \mid U' \mid)$ clauses and $O(k * (\mid O' \mid + \mid U' \mid))$ variables.

To complete the lifting transformation, we need to give the schemas for the reduction of lifted SAT to SAT. The 5 schemas in Figure 6 are same as those in [Kautz *et. al.* 96]. Here $t, u, w, f(t_1, t_2, t_3, ..., t_k), f(u_1, u_2, ..., u_k)$ denote the terms from the lifted version.

1. $\wedge_{i=0}^{k-1} \wedge_{j=1}^{|O'|} (o'_j(i) \Rightarrow (\wedge_{s=1}^{R_j} n'_{js}(i)))$

2. $\wedge_{i=0}^{k-1} \wedge_{j=1}^{|O'|} (o'_j(i) \Rightarrow$
$((\wedge_{j_1=1}^{A_j} a'_{jj_1}(i+1)) \wedge (\wedge_{j_2=1}^{D_j} \neg d'_{jj_2}(i+1))))$

3. $\wedge_{i=0}^{k-1} \wedge_{j=1}^{|U'|} ((u'_j(i) \wedge \neg u'_j(i+1)) \Rightarrow$
$(\vee_{s=1, Can\_Del(o'_s, u'_j)}^{|O'|} o'_s(i)))$

$\wedge_{i=0}^{k-1} \wedge_{j=1}^{|U'|} ((\neg u'_j(i) \wedge u'_j(i+1)) \Rightarrow$
$(\vee_{s=1, Can\_Add(o'_s, u'_j)}^{|O'|} o'_s(i)))$

4. $\wedge_{i=1}^{h} a_i(k)$

5. $(\wedge_{i=1}^{|I|} l_i(0)) \wedge (\wedge_{j=1, u_j \notin I}^{|U|} \neg u_j(0))$

6. $\wedge_{i=1}^{|V|} (\vee_{j=1}^{|Dom_i|} (x_i = c_{ij}))$

Figure 5: Lifted version of ground state-based planning

1. $t = t$
2. $(t = u) \Rightarrow (u = t)$
3. $((t = u) \wedge (u = w)) \Rightarrow (t = w)$
4. $(f(t_1, t_2, ..., t_k) = f(u_1, u_2, ..., u_k)) \Leftrightarrow$
$((t_1 = u_1) \wedge (t_2 = u_2) \wedge .. \wedge (t_k = u_k))$
5. $\neg(t = u), t, u$ clash.

Figure 6: Additional clauses for reduction from lifted SAT to SAT

Since the lifted version of the ground state-based encoding with explanatory frame axioms is strictly smaller than the lifted version of the smallest causal encoding and since the reduction from lifted SAT to SAT in the causal encoding cannot be smaller than the corresponding size for the state-based encoding, we have the theorem:

Theorem 2. *The lifted state-based encoding with explanatory frame axioms is strictly smaller than any lifted causal encoding.*

6 Empirical Evaluation

Until now, we have shown the dominance of various types of encodings in terms of the asymptotic sizes (in terms of number of variables and clauses). Ultimately of course, we are more interested in how the encodings behave in practice. There are two possible reasons why the practice may deviate from the theory. First, the asymptotic analyses miss the constant factors, and actual encodings may in fact be larger because of the relative sizes of these ignored constants. Second, and perhaps more important, the correlation between the size of a SAT encoding and the hardness of solving it is by no means perfect. Indeed, it is known that adding certain types of constraints (including mutual exclusion constraints, domain specific constraints etc.) while increasing the encoding size, wind up facilitating simplification (through techniques such as unit propagation), making the encodings much easier to solve.

To verify if the size-based dominances that we have discussed in this paper are correlated with the hardness of solving the encodings, we conducted empirical comparisons among the causal encoding developed by [Kautz *et. al.* 96]

Domain	State-based			Our best Causal encoding (Sec. 3.3)			Kautz et. al.'s Causal encoding		
(Steps)	#Vars	#Clauses	Time	# Vars	#Clauses	Time	#Vars	#Clauses	Time
Ferry (15)	390	1519	0.23	855	4144	1.01	4714	58444	81.29
Ferry (19)	588	2436	4.17	1291	7224	125.16	8535	138172	-
Ferry (23)	826	3615	48.54	1815	11504	-	13988	280328	-
Tsp (8)	217	553	0.02	497	1661	0.07	1809	10825	2.11
Tsp(14)	631	1640	0.06	1457	6770	0.88	7785	88873	2.42
Tsp(20)	1199	3138	0.17	2779	16308	6.58	19618	335818	-
Log(19)	921	2639	0.13	2004	12120	-	13051	211696	-
Log(12)	378	1068	0.04	822	3636	0.63	3803	36611	165.97

Figure 7: Empirical results on the performance of selected encodings. Times are in CPU seconds. A "-" indicates that the encoding was not solved within 5 minutes of CPU time on a Sun Ultra with 128M RAM.

(see Section 3.1), the causal encoding that we found to be the smallest based on our analysis of the spectrum of encodings (see Section 3.3), as well as the best state-based encoding (those with explanatory frame axioms; see Section 2.2). Our experiments involved encoding a specific planning problem in each of these encodings. Following the practice of [Kautz & Selman, 96], the number of steps we used in the encodings were greater than or equal to the minimal length solution for the problem (thus eliminating the need for solving encodings of various lengths). Each of the encodings were solved with the SATZ solver[1], a state-of-the-art systematic SAT solver.

The results of our empirical study are shown in Figure 7. The descriptions of the benchmark domains we used are available at *www.cs.yale.edu/HTML/YALE/CS* in the directory *HyPlans/mcdermott.html*. "Tsp" denotes the traveling sales person domain, while "ferry" denotes the ferry domain involving transportation of objects. "Log" denotes the logistics domain. The number of steps in the encodings were same as the number of actions in the plans. Though many of the irrelevant actions were eliminated from consideration before generating the encodings, the same actions were used in all encodings of each problem.

The results show that our improved causal encoding (from section 3.3) could be solved significantly faster than the causal encoding of [Kautz *et. al.* 96]. They also show that the state-based encoding with explanatory frame axioms was still the fastest to solve. The encoding sizes, in terms of number of variables and clauses, are in accordance with the asymptotic relations. We also repeated the experiments where the encodings were first processed with traditional simplifiers (e.g. unit propagation), before being solved. The simplification did not have any appreciable effect on the relative performances of the three encodings.

7 Related Work & Discussion

As we noted, plan-space encodings are based on the ideas of proving the correctness of a plan in terms of establishment and declobbering of all goals and action preconditions in a plan. Historically, these ideas were associated with partial order planning [McAllester & Rosenblitt, 91; Penberthy & Weld, 92]. Partial order planning is known to be a more flexible and efficient form of plan synthesis [Barrett & Weld, 94], and this was to some extent the motivation for the initial interest in the causal encodings. Given this background, the results of this paper seem paradoxical, in as much as they show that causal (plan-space) encodings are dominated by the state-based encodings.

Upon closer examination however, this apparent paradox turns out to be an artifact of a misunderstanding of the relation between traditional planning algorithms, and the SAT encodings inspired by those algorithms. The primary difference between state-space and plan-space (partial-order) planners is the specific way a partial plan is extended – state space planners extend the suffix or the prefix of the plan, while partial order planners have the flexibility to insert steps anywhere in the partial plan. The specific strategies used to check if the plan under consideration constitutes a solution are in fact interchangeable [Kambhampati 97].

In contrast, as we have seen throughout this paper, the various causal encodings are distinguished by the various ways of proving the correctness of a plan. The issues of (partial) plan extension are irrelevant for SAT encodings, since SAT-based planning in essence starts with a fixed length disjunctive structure, and checks to see if some conjunctive substructure of it corresponds to a valid plan for the problem. It is only because extension is irrelevant that we were able to consider replacing partial ordering with contiguous ordering (which ultimately resulted in a better plan-space encoding).

From the above perspective, there is no reason to expect that the advantages of partial-order planners over state space planners, which are based largely on the flexibility of inserting steps anywhere in the partial plan, will transfer over to plan space (causal) encodings and state-based encodings that are distinguished by the differences in proof strategies. In fact, since causal proofs consider establishment and declobbering for each precondition of each step separately, they are an inefficient way of checking the correctness of a given action sequence. The reason they are used in partial order planners is that such planners need to interleave refinement and correctness checking of partial plans, and since plan-space refinements add actions without fixing their absolute position, causal proof strategies provide the best means of incrementalizing (finite-differencing) the proof attempts. This flexibility is clearly irrelevant in solving SAT encodings.

7.1 Two uses for Causal encodings

Although causal encodings do not have any advantages in the standard STRIPS-planning tasks, we now show that they

[1] available from *aida.intellektik.informatik.th-darmstadt.de* in *~hoos/SATLIB*

could be advantageous in incremental planning scenarios as well as in exploiting causal domain knowledge.

Incremental planning: SAT-based planning has hither-to concentrated on "from-scratch" planning scenarios–where the planner is presented with just the specification of the planning problem. An equally important problem, that has been considered in the traditional planning scenarios, is that of "incremental planning" that arises in the context of replanning and plan-reuse. In this case, in addition to a problem specification, one is given a partial plan, with the requirement that as much of that plan as possible be reused in solving the new problem. Solving such problems could potentially benefit from the ability to insert steps flexibly into the given plan [Ihrig & Kambhampati, 94]. For example, consider a scenario where we are reusing a 2 step plan $[o_2 o_1]$ to solve a new problem, and suppose there is a solution to the new problem that involves inserting a new action o_3 at an arbitrary place in the current plan. If we solved the original problem using a causal encoding (with partial ordering), then it would be feasible to solve the new problem by incrementally extending the original encoding and re-solving it. If we want to keep the original steps as part of the new plan, we need only change their step-action mapping axiom appropriately, e.g. $(p_1 = o_1)$ (or $((p_1 = o_1) \vee (p_1 = \phi))$ if we want to allow removal of old actions.)

In contrast, if the original problem is to be solved with a state-based encoding, one has to either (i) represent a disjunction of all possible ways of respecting the the constraints from the old plan, e.g. $((o_2(0) \wedge o_1(1)) \vee (o_2(0) \wedge o_1(2)) \vee (o_2(1) \wedge o_1(2)))$ or (ii) make multiple copies of the old plan (only if the actions from the old plan may need to be reordered or removed), and reserve multiple places for the inclusion of new actions, e.g. $((o_3(0) \vee \phi(0)) \wedge o_2(1) \wedge (o_3(2) \vee \phi(2)) \wedge o_1(3) \wedge (o_3(4) \vee \phi(4)))$. In addition to increasing the size of the encodings, this approach unfortunately also opens up the possibility of having redundant occurrences of o_3 in the final plan. Indeed in the case of plan merging and reuse, it was found that the causal encodings of some problems were smaller and faster to solve than the state-based encodings [Mali 99(b)].

Using causal domain knowledge: Another scenario where the causal encodings were found to be smaller and faster to solve than the state-based encodings is the hierarchical task network planning problem cast as satisfiability [Mali 99(a)]. The causal encodings naturally capture the precedence constraints and causal links specified in the task reduction schemas. On the other hand, in the state-based encodings, these constraints need to be represented as the disjunction of all total orders on the steps that are consistent with the partial order in the constraints from the reduction schemas. For example, in a k step encoding, the link $o_2 \xrightarrow{f} o_3$ needs to be represented as $\vee_{i=0}^{k-1} \vee_{j=i+1}^{k} (o_2(i) \wedge o_3(j) \wedge (\wedge_{q=i+1}^{j} f(q)))$. This approach significantly increases the encoding size [Mali 99(a)].

8 Conclusion

In this paper we provided a systematic analytical and empirical comparison of plan-space (causal) encodings and state-based encodings. We pointed out that the two types of encodings differ mainly in the way they attempt to prove the correctness of a plan. We then showed that there can be a large variety of causal encodings corresponding to different ways of carrying out causal proofs. The critical dimensions are the specific ways in which establishment and declobbering of pre-conditions is ensured, and the type of ordering assumed between the steps of the encoding. We showed that the causal encoding that was previously studied in the literature corresponds to one specific variation, and presented two other variations that are significantly smaller. We went on to show that even our smallest causal encodings cannot be smaller in size than the smallest state-based encodings. We also showed that the "lifting" transformation does not affect this relation. We bolstered our claims by presenting empirical results that demonstrate that the relative encoding sizes are indeed correlated with the hardness of solving them. Finally, we discussed why it should not be surprising that the primacy of traditional plan-space planners over state-space planners does not carry over to their respective SAT encodings, and showed that causal encodings might have advantages in solving incremental planning problems.

References

[Barrett & Weld 94] Anthony Barrett and Daniel Weld, Partial order planning: Evaluating the possible efficiency gains, Artificial Intelligence 67: 71-112, May 1994.

[Chapman 87] David Chapman, Planning for conjunctive goals, Artificial Intelligence, Vol. 32, No. 3, July 1987.

[Ernst *et. al.* 97] Michael Ernst, Todd Millstein and Daniel Weld, Automatic SAT compilation of planning problems, Proccedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.

[Ihrig & Kambhampati 94] Laurie Ihrig and Subbarao Kambhampati, "Derivation Replay for Partial-order Planning," Proc. 12th Natl. Conf. on Artificial Intelligence (AAAI), August 1994.

[Kambhampati *et. al.* 95] Subbarao Kambhampati, Craig Knoblock and Qiang Yang. Planning as Refinement Search: A unifying framework for evaluating design tradeoffs in partial order planning. *Artificial Intelligence.* Vol. 76. No. 1-2, September 1995. pp. 167-238.

[Kambhampati 97] Subbarao Kambhampati, Refinement planning as a unifying framework for plan synthesis, AI Magazine, 18(2): 67-97, Summer 1997.

[Kautz *et. al.* 96] Henry Kautz, David McAllester and Bart Selman, Encoding plans in propositional logic, Proc. of Knowledge Representation and Reasoning Conference, (KRR), 1996.

[Kautz & Selman 96] Henry Kautz and Bart Selman, Pushing the envelope: Planning, Propositional logic and Stochastic search, Proc. of the National Conference on Artificial Intelligence (AAAI), 1996.

[Mali 99(a)] Amol D. Mali, Hierarchical task network planning as satisfiability, Ph.D thesis, Dept. of computer science & engg., Arizona state university, April 1999.

[Mali 99(b)] Amol D. Mali, Plan merging and plan reuse as satisfiability, Technical report TR-99-01, Dept. of computer science & engg., Arizona state university, 1999.

[McAllester & Rosenblitt 91] David McAllester and David Rosenblitt, Systematic non-linear planning, Proceedings of the National Conference on Artificial Intelligence (AAAI), 1991.

[Minton *et. al.* 91] S. Minton, J. Bresina and M. Drummond, Commitment strategies in planning: a comparative analysis, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1991.

[Penberthy & Weld 92] J. S. Penberthy and D. Weld, UCPOP: A sound, complete partial order planner for ADL, Proceedings of Knowledge Representation & Reasoning (KRR), 1992.

[Selman *et. al.* 97] Bart Selman, Henry Kautz and David McAllester, Ten challenges in propositional reasoning and search, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI), 1997.