

LOCALIZED TEMPORAL REASONING USING SUBGOALS AND ABSTRACT EVENTS

SHIEU-HONG LIN, THOMAS DEAN¹

Department of Computer Science, Brown University, Providence, RI 02912

We are concerned with temporal reasoning problems where there is uncertainty about the order in which events occur. The task of temporal reasoning is to derive an event sequence consistent with a given set of ordering constraints to achieve a goal. Previous research shows that the associated decision problems are hard even for very restricted cases. In this paper, we investigate locality in event ordering and causal dependencies. We present a localized temporal reasoning algorithm that uses subgoals and abstract events to exploit locality. The computational efficiency of our algorithm for a problem instance is quantified by the inherent locality in the instance. We theoretically demonstrate the substantial improvement in performance gained by exploiting locality. This work provides a solid evidence of the usefulness of localized reasoning in exploiting locality.

Key words: Dynamical Systems, Temporal Reasoning, Planning, Locality, Localized Reasoning, Subgoals, Abstract Events

1. INTRODUCTION

We are interested in a dynamical system modeled by (1) a set of conditions \mathcal{P} , (2) a set of events \mathcal{E} , and (3) a set of ordering constraints \mathcal{O} on \mathcal{E} . \mathcal{P} , \mathcal{E} , and \mathcal{O} are all finite sets. Each condition is either true or false at every point in time. The truth values of the conditions in \mathcal{P} at a point in time determine the system state at that time. In particular, the initial values of conditions determine the initial state of the dynamical system. Each event in \mathcal{E} occurs exactly once and changes the values of some conditions instantaneously. The system state persists during the period of time between two successive occurrences of events. When an event e occurs, e precipitates a change of system state according to (1) the system state immediately before e occurs and (2) a set of *causal rules* associated with e . A causal rule associated with an event is a STRIPS-like operator representing a possible pair of the precondition and effect of the event; the operator is applicable only when the precondition is true immediately before the event occurs. An event changes the system state according to exactly one of the applicable rules; ties are broken arbitrarily when two or more rules are applicable at a time. In this way, each event determines a state-transition relation on the state space determined by \mathcal{P} [LD93].

We assume that each event in \mathcal{E} occurs exactly once. If necessary, we can represent multiple occurrences of an event at distinct points in time by (1) replicating the event a specified number of times, and (2) considering each replica of the event as a distinct event. The order in which the events occur is partially specified by the ordering constraints in \mathcal{O} . An event e can occur at a point in time if and only if (1) at least one of the rules associated with e is applicable at that time, and (2) none of the ordering constraints in \mathcal{O} is violated if e occurs at that time. The events in \mathcal{E} occur one after another in an order consistent with the ordering constraints in \mathcal{O} . The dynamical system evolves through a finite number of state transitions as the

¹This work was supported in part by a National Science Foundation Presidential Young Investigator Award IRI-8957601, by the Advanced Research Projects Agency of the Department of Defense monitored by the Air Force under Contract No. F30602-91-C-0041, and by the National Science foundation in conjunction with the Advanced Research Projects Agency of the Department of Defense under Contract No. IRI-8905436. The first author can be contacted at shl@cs.brown.edu or (401) 863-7668.

events in \mathcal{E} occur as a sequence.

The dynamical system can evolve in many different ways due to the uncertainty in (1) the order in which the events occur and (2) the choices of applicable causal rules. It is important to predict and control the evolution of such a dynamical system. In particular, we would like to (1) specify a set of goal states (or undesirable states), (2) detect the possibility of reaching a goal state (or an undesirable state), and (3) report as a solution (or a warning) such a sequence of events and the choices of applicable rules that lead to a goal state (or an undesirable state) if this is possible. This is the task of *temporal projection* investigated in this paper. Temporal projection can be considered as a special kind of planning. In temporal projection, we do not have full control over events. Instead, we need to achieve a goal under the restrictions that (1) only a fixed number of events can be used, (2) each event occurs once, and (3) the events must occur in a total order consistent with the ordering constraints in \mathcal{O} .

The associated decision problems for planning [BK91] [Byl91] [Cha87] [DB88] [GN91] and temporal projection [DB88] [NB92] [LD93] are hard even for very restricted cases. Dean and Boddy [DB88] study a variant of temporal projection. They show that temporal projection is NP-Complete even if we severely restrict the form of causal rules and the number of rules associated with events. Nebel and Bäckström [NB92] describe a situation in which (1) a restricted case of temporal projection is NP-Complete while (2) the corresponding planning problem can be solved in polynomial time [NB92] if there are no ordering constraints on events and we allow an event to occur an arbitrary number of times. We have been trying to understand why temporal projection problems are so difficult and what, if any, structure might be extracted to expedite decision making and inference [LD93]. We identify (1) the kinds of events, (2) the ordering constraints on events, and (3) the size of the state space, as three important factors influencing computational complexity. Our complexity results indicate that we have very limited ability to consider a large number of conditions that result in a huge state space. To expedite temporal projection, we exploit locality in event ordering and the dependencies among conditions and events. This allows us to focus on small subsets of the set of conditions \mathcal{P} separately rather than considering the entire set \mathcal{P} at a time.

The locality in event ordering in a particular problem instance is modeled by a hierarchy of regions. A region is a collection of closely related events which occur as an atomic group. An event outside a region must occur either before or after the events in the region. For each region, we can identify four characteristic subsets of the set of conditions \mathcal{P} , which reflect locality in causal dependencies. The locality in a problem instance can be quantified by the maximum size of these characteristic subsets of \mathcal{P} , and the maximum branching factor of the region hierarchy. To exploit locality, we develop a localized algorithm for temporal projection. The notions of subgoals and abstract events play important roles in our localized algorithm. Given a problem instance, the goal is decomposed into subgoals in individual regions. In each region, we construct a local search space and conduct a local search to identify the possible preconditions and effects that are critical in achieving the subgoal in that region. We compile these possible preconditions and effects as a set of causal rules, and refer to them as the abstract event associated with the region. Starting from the bottom level of the region hierarchy, we search the local search spaces of the regions level by level. As soon as we reach the root, we are able to generate a solution according to the information associated with the derived abstract events. In a local search, we only consider the conditions appearing in the characteristic subsets of \mathcal{P} associated with the corresponding region. The locality embedded in a problem instance allows us to

construct smaller local search spaces. The computational complexity of our algorithm is determined by the locality of a problem instance. We theoretically demonstrate the substantial performance improvement gained by exploiting locality.

The remainder of the paper is organized as follows. In Section 2, we use an example to illustrate the application of temporal projection. In Section 3, we formally describe the temporal projection problem and its variants. Section 4 investigates the complexity trade-offs of temporal projection. Section 5 illustrates the notions of regions organized as a region hierarchy. In Section 6, we depict locality in causal dependencies among regions. For each region, we identify the characteristic condition subsets and define the subgoals and abstract events of the region. We propose our localized algorithm in Section 7, and analyze its computational complexity in Section 8. Section 9 provides additional pointers to related work.

2. THE BATHROOM SHARING PROBLEM

Consider the following example. In Figure 1-a, six tenants, $e_1, e_2, e_3, e_4, e_5, e_6$ share a bathroom in their apartment. $\{e_1, e_2\}$ represent couple X . $\{e_3, e_4\}$ represent couple Y . $\{e_5, e_6\}$ represent couple Z . We refer to these six tenants as a group W . In the bathroom, there are eight switches (a, b, c, d, e, f, g, h).

All six tenants work late at night. When they arrive home, each of them uses the bathroom once before going to sleep. While using the bathroom, they may turn on or turn off the switches. Figure 1-b depicts their individual behaviors in the bathroom. For example, e_1 likes to turn on a and b and turn off c if a and b are both off when he enters the bathroom; otherwise, e_1 just lets all switches remain as they were when he entered.

Figure 1-c displays the ordering constraints regarding the use of bathroom. For each couple, they use the bathroom in a row. In other words, immediately after one uses the bathroom, the other one will use the bathroom. Couple X always use the bathroom after couple Y use it. Concerning couple Z , e_6 always uses the bathroom before e_5 uses it. The landlord of the apartment uses the bathroom and goes to sleep before the tenants come back. After using the bathroom, the landlord turns off all switches. The following scenarios describe various temporal reasoning tasks concerned with the consequences of the tenants' behavior.

Scenario 1: The landlord is informed that due to a design fault the whole electrical system may burn out if the switches a, b, d, e, g, h all remain turned on overnight. The landlord would like to

- (1) determine if all of the switches a, b, d, e, g, h could be turned on overnight after all tenants use the bathroom, and, if this is possible,
- (2) produce a total order so that the undesirable situation does occur, and
- (3) explain why the total order results in the undesirable situation.

Scenario 2: To save electricity, the landlord would like to derive a total order on the tenants' use of the bathroom such that all switches are turned off after the last tenant leaves the bathroom.

Scenario 3: The landlord is informed that the whole electrical system may burn out as soon as the switches a, b, d, e, g, h are on at the same time. The landlord would like to determine whether such an undesirable situation can occur while one of the

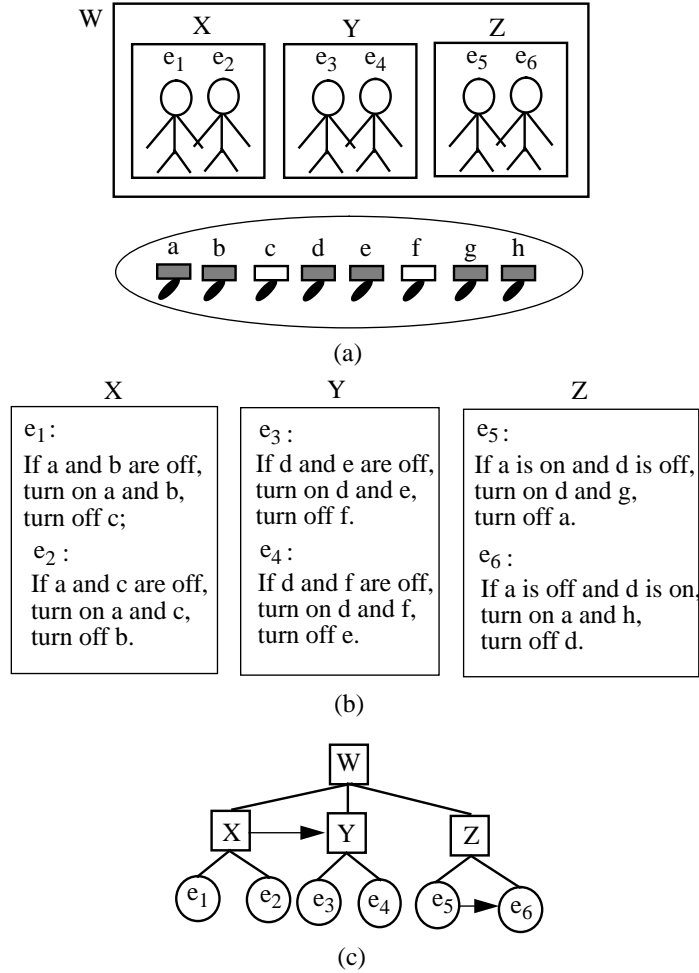


FIGURE 1. Temporal reasoning about sharing a bathroom

tenants is using the bathroom.

Scenario 4: Tenant e_5 is informed that the electrical system may burn out as soon as the switches a, b, d, e, g, h are on at a time. Tenant e_5 would like to determine whether such an undesirable situation can occur while e_5 is using the bathroom.

The bathroom example describes a partial plan for using the bathroom. In the following, we illustrate the dynamical system represented by this partial plan and describe the application of temporal projection in the four scenarios.

States: We have eight conditions, each of which describes whether a particular switch is on or off. A condition is true (false) when the corresponding switch is off (on). The values of the eight conditions determine the state of the dynamical system. In the initial state, all conditions are true since all switches are off immediately before

the tenants come home.

Events: The use of the bathroom by a tenant is an event, which can change the system state according to the tenant’s behavior as described in Figure 1-b. Therefore the six tenants corresponds to six events that affect the system state. In this example, each event is described by an if-then sentence while in general an event may be associated with multiple causal rules to describe its conditional effects.

Uncertainty in the system’s evolution: There is uncertainty in the order in which the events occur. Many total orders on these six events are consistent with the ordering constraints depicted in Figure 1-c. The actual order in which the events occur critically affects the system’s evolution.

The task of temporal projection: In Scenarios 1, 3, and 4, an undesirable state is a state in which the switches a, b, d, e, g, h are on and the corresponding conditions are false. The landlord or the tenants can use temporal projection to (1) predict the possibility of reaching an undesirable state, and also (2) derive an explanation of how such a state might be reached. In Scenario 2, a goal state is a state in which all switches are off and all conditions are true. The landlord can apply temporal projection to derive a total order on the events which ends in a goal state. This allows him to provide a schedule to the tenants for using the bathroom such that all switches are turned off at the end.

3. TEMPORAL PROJECTION

3.1. Terminology and Representation

First of all, we introduce the following terminology and/or representation regarding a dynamical system $\langle \mathcal{P}, \mathcal{E}, \mathcal{O} \rangle$ where (1) $\mathcal{P} = \{p_1, p_2, \dots, p_l\}$ is a set of conditions, (2) $\mathcal{E} = \{e_1, e_2, \dots, e_n\}$ is a set of events, and (3) \mathcal{O} is a set of ordering constraints on \mathcal{E} .

Definition 1. (States and the State Space) The set of conditions \mathcal{P} determines a state space $S_{\mathcal{P}} = \{true, false\}^l$. A state in $S_{\mathcal{P}}$ is represented as a vector $\langle v_1, v_2, \dots, v_l \rangle$ where v_i indicates the value of condition p_i at a point in time.

Definition 2. (Expression) An expression is a set of condition/value pairs of the form $\langle p_i, v_i \rangle$ where p_i is a condition and v_i is either *true* or *false*. An expression α is true in a state u if and only if for each condition/value pair $\langle p_i, true \rangle$ ($\langle p_j, false \rangle$) in α p_i is true in u (p_j is false in u). An empty expression $()$ is true in every state.

Definition 3. (Causal Rules) A causal rule r is represented as $\alpha \rightarrow \beta$ where α and β are expressions. α is the precondition and β is the postcondition or effect of rule r , which is applicable in state u if and only if α is true in state u . Rule r can map state u to state v if and only if (1) rule r is applicable in state u , (2) β is true in state v , and (3) the values of the conditions not appearing in β remain the same in v as they are in u .

Definition 4. (Events) An event e is represented as a set of causal rules. Event e can cause a state transition from state u to state v if and only if event e can map

state u to state v by applying exactly one of the applicable rules; a tie is broken arbitrarily if two or more rules of event e are applicable in state u .

Definition 5. (Possible Event Sequences) A possible event sequence \mathbf{q} of length h over a set of events \mathcal{E} ($h \leq n$) is a sequence $\langle e_1, e_2, \dots, e_h \rangle$ where (1) e_1, \dots, e_h are distinct events in \mathcal{E} and (2) the order in which the events appear in \mathbf{q} is consistent with the ordering constraints in \mathcal{E} . We define the following sets of event sequences:

$$\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}} = \{ \mathbf{q} \mid \mathbf{q} \text{ is a possible event sequence of length } n \},$$

$$\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}} = \{ \mathbf{q} \mid \mathbf{q} \text{ is a possible event sequence of length equal to or less than } n \}, \text{ and}$$

$$\overline{\mathbf{Q}}_{\mathcal{O},e}^{\mathcal{E}} = \{ \mathbf{q} \mid \mathbf{q} \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}} \text{ and event } e \text{ is the tail of } \mathbf{q} \}.$$

Definition 6. (Trajectory) A sequence of states $\langle s_0, s_1, s_2, \dots, s_h \rangle$ is a trajectory of an event sequence $\mathbf{q} = \langle e_1, e_2, \dots, e_h \rangle$ if and only if each event e_i can cause a state transition from state s_{i-1} to state s_i for $1 \leq i \leq h$. The trajectory of a possible event sequence provides a possible trace of the system's evolution.

3.2. The Temporal Projection Problem

An instance of the *temporal projection problem* is defined by $\langle \mathcal{P}, \mathcal{E}, \mathcal{O}, s_0, \mathcal{G} \rangle$ where

- \mathcal{P} is a finite set of conditions;
- \mathcal{E} is a finite set of events;
- \mathcal{O} is a finite set of ordering constraints on \mathcal{E} ;
- \mathcal{P} , \mathcal{E} , and \mathcal{O} together model a dynamical system;
- s_0 is the initial state, $s_0 \in S_{\mathcal{P}}$;
- \mathcal{G} is an expression and \mathcal{G} specifies a set of *goal states* in $S_{\mathcal{P}}$ in which \mathcal{G} is true.

Definition 7. (The Tasks of Temporal Projection) Given an instance of the temporal projection problem, the PRJ task, the PRJ1 task, and the PRJ2 task of temporal projection are to determine the existence of an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$, $\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$, and $\overline{\mathbf{Q}}_{\mathcal{O},e}^{\mathcal{E}}$ respectively such that (1) the events can occur as the sequence \mathbf{q} and (2) the dynamical system may enter a goal state immediately following \mathbf{q} . In addition, we generate such an event sequence \mathbf{q} if one exists, and determine the rules applied by the individual events to enter a goal state.

Theorem 1. Given an instance of the temporal projection problem, the PRJ task, the PRJ1 task, and the PRJ2 task can be reduced to one another by transforming the problem instance. All these transformations can be done in polynomial time by (1) adding at most two more conditions and two more events, and (2) appropriately modifying the ordering constraints and causal rules.²

Throughout this paper, we consider the PRJ task³ as the standard form of temporal projection. The localized algorithm and its analysis in this paper focus on the PRJ task, which are also applicable to the PRJ1 task and the PRJ2 task after transforming the problem instances accordingly.

²The proofs for all theorems stated in this paper are provided in Appendix A at the end of this document.

³The PRJ1 task is considered by Lin and Dean in [LD93] and the PRJ2 task is considered by Dean and Boddy in [DB88] and Nebel and Bäckström in [NB92].

$$\begin{aligned}
 e_1 &= \{(\langle a, \text{true} \rangle \langle b, \text{true} \rangle) \rightarrow (\langle a, \text{false} \rangle \langle b, \text{false} \rangle \langle c, \text{true} \rangle) \\
 &\quad (\langle a, \text{false} \rangle) \rightarrow (), \quad (\langle b, \text{false} \rangle) \rightarrow ()\} \\
 e_2 &= \{(\langle a, \text{true} \rangle \langle c, \text{true} \rangle) \rightarrow (\langle a, \text{false} \rangle \langle c, \text{false} \rangle \langle b, \text{true} \rangle) \\
 &\quad (\langle a, \text{false} \rangle) \rightarrow (), \quad (\langle c, \text{false} \rangle) \rightarrow ()\} \\
 e_3 &= \{(\langle d, \text{true} \rangle \langle e, \text{true} \rangle) \rightarrow (\langle d, \text{false} \rangle \langle e, \text{false} \rangle \langle f, \text{true} \rangle) \\
 &\quad (\langle d, \text{false} \rangle) \rightarrow (), \quad (\langle e, \text{false} \rangle) \rightarrow ()\} \\
 e_4 &= \{(\langle d, \text{true} \rangle \langle f, \text{true} \rangle) \rightarrow (\langle d, \text{false} \rangle \langle f, \text{false} \rangle \langle e, \text{true} \rangle) \\
 &\quad (\langle d, \text{false} \rangle) \rightarrow (), \quad (\langle f, \text{false} \rangle) \rightarrow ()\} \\
 e_5 &= \{(\langle a, \text{false} \rangle \langle d, \text{true} \rangle) \rightarrow (\langle a, \text{true} \rangle \langle d, \text{false} \rangle \langle g, \text{false} \rangle) \\
 &\quad (\langle a, \text{true} \rangle) \rightarrow (), \quad (\langle d, \text{false} \rangle) \rightarrow ()\} \\
 e_6 &= \{(\langle a, \text{true} \rangle \langle d, \text{false} \rangle) \rightarrow (\langle a, \text{false} \rangle \langle d, \text{true} \rangle \langle h, \text{false} \rangle) \\
 &\quad (\langle a, \text{false} \rangle) \rightarrow (), \quad (\langle d, \text{true} \rangle) \rightarrow ()\}
 \end{aligned}$$

FIGURE 2. The events and causal rules in the bathroom example

Example 1. Scenarios 1 and 2 in Section 2.1 are examples of the PRJ task while Scenarios 3 and 4 are examples of the PRJ1 task and the PRJ2 task respectively. In the following, we use Scenario 1 in the bathroom example to provide a detailed description of an instance of the temporal projection problem.

- $\mathcal{P} = \{a, b, c, d, e, f, g, h\}$, which corresponds to the eight switches in the bathroom. A condition is true if and only if the corresponding switch is off.
- $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$. Each event corresponds to the use of the bathroom by a particular tenant. Figure 2 display the causal rules associated with the individual events.
- Figure 1-c depicts the ordering constraints \mathcal{O} on \mathcal{E} . For convenience, we also refer to \mathcal{E} as W . The events in the three event subsets X , Y , and Z must occur as three atomic groups, where $X = \{e_1, e_2\}$, $Y = \{e_3, e_4\}$, $Z = \{e_5, e_6\}$. The events in Y must occur before the events in X . Event e_6 must occur before event e_5 .
- In the initial state s_0 , all conditions are initially true, since the landlord turns off all switches before the tenants use the bathroom.
- $\mathcal{G} = (\langle a, \text{false} \rangle \langle b, \text{false} \rangle \langle d, \text{false} \rangle \langle e, \text{false} \rangle \langle g, \text{false} \rangle \langle h, \text{false} \rangle)$.
- Our task is to (1) determine the existence of an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ that may end in a goal state, in which the goal \mathcal{G} is true, and (2) if such a \mathbf{q} exists, generate \mathbf{q} and determine the rules applied by the individual events to enter a goal state.

4. THE COMPUTATIONAL COMPLEXITY OF TEMPORAL PROJECTION

Given an instance of the temporal projection problem, we can first guess a solution event sequence \mathbf{q} and the rules applied by the individual events. We can then verify in polynomial time whether a goal state is reached immediately following \mathbf{q} . Therefore we have the following lemma.

Lemma 1. The temporal projection problem is in NP.

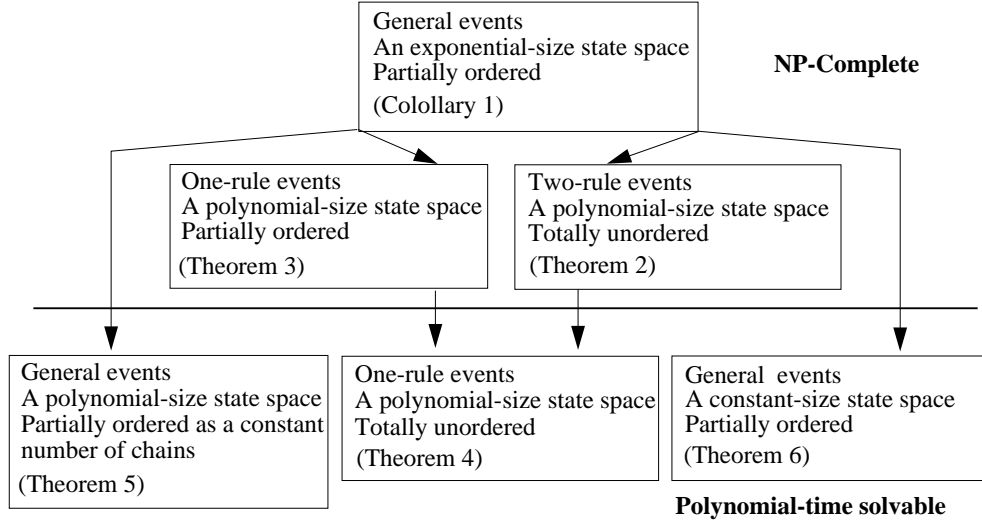


FIGURE 3. Complexity trade-offs of temporal projection

In the following, we show that (1) the kinds of events, (2) the ordering constraints on events, and (3) the size of the state space all contribute to the computational complexity. First we define the following terms.

Definition 8. (One-rule Events and Two-rule Events) A one-rule event is associated with a single rule that is applicable in exactly one state and maps that state to a distinct state. A two-rule event is associated with two rules, each of which is applicable in exactly one state and maps that state to a distinct state.

Remark. In general, an event can have multiple rules, each of which can map many states to states other than themselves.

Definition 9. (Partially Ordered as Chains and Totally Unordered) Given a set of events \mathcal{E} , we say that \mathcal{E} is partially ordered if \mathcal{E} is constrained by an arbitrary partial order $<$; we say that \mathcal{E} is partially ordered as m chains if (1) \mathcal{E} can be partitioned into m disjoint event subsets, (2) the events in each subset are totally ordered as a chain, and (3) the events in the entire set \mathcal{E} can only be ordered by interleaving these m chains; we say that \mathcal{E} is totally unordered if every total order on the set of events \mathcal{E} is possible.

Definition 10. (The Size of the State Space) Given a set of events \mathcal{E} and a set of conditions \mathcal{P} , we say that the state space $S_{\mathcal{P}}$ is of constant size if \mathcal{P} is composed of $O(1)$ number of conditions; we say that the state space is of polynomial size if \mathcal{P} is composed of $O(\log|\mathcal{E}|)$ number of conditions; and we say that the state space is of exponential size if \mathcal{P} is composed of $O(|\mathcal{E}|)$ number of condition.

Figure 3 displays the complexity trade-offs regarding events, ordering constraints, and state spaces, where

- events are categorized as either one-rule events, two-rule events or general events according to how they trigger state transitions;
- events can be either (1) partially ordered as a constant number of chains, (2) totally unordered, or (3) constrained by an arbitrary partial order;
- a state space can be either constant-size, polynomial-size, or exponential-size with respect to the number of events.

When there is no structure in event ordering, the following two theorems indicate that we have very limited ability to deal with a polynomial-size state space determined by $O(\log |\mathcal{E}|)$ number of conditions.

Theorem 2. Temporal projection regarding two-rule events, totally unordered, with a polynomial-size state space is NP-Complete.

Theorem 3. Temporal projection regarding one-rule events with an arbitrary partial order and a polynomial-size state space is NP-Complete.⁴

Corollary 1. Temporal projection regarding general events with an arbitrary partial order and an exponential-size state space is NP-Complete.

The following three theorems demonstrate that temporal projection is more tractable when we have (1) available structure in causal rules or event ordering or (2) a small state space. These complexity results motivate our effort to exploit inherent locality in event ordering and the dependencies among conditions and events.

Theorem 4. Temporal projection regarding one-rule events, totally unordered, with a polynomial-size state space is solvable in polynomial time.

Theorem 5. Temporal projection regarding general events partially ordered as $O(1)$ number of chains of length $O(|\mathcal{E}|)$ or $O(\log |\mathcal{E}|)$ number of chains of $O(1)$ length with a polynomial-size state space can be solved in polynomial time.

Theorem 6. Temporal projection regarding general events with an arbitrary partial order and a constant-size state space is solvable in polynomial time.

5. LOCALITY IN EVENT ORDERING

When totally unordered, events can occur in an arbitrary order. However, there is locality in event ordering if the occurrences of events or groups of events closely relate to one another. Several closely related events may always occur as an atomic group; an event not in the group either occurs before or after the group. Similarly, several atomic groups may always occur as a macro atomic group. In this way, locality in event ordering can be introduced at many levels. For example, the events corresponding to the three couples, X , Y and Z respectively in the bathroom example exhibit this kind of temporal locality. In the remainder of this paper, we investigate locality in event ordering that can be modeled by a *region hierarchy* and *regional ordering constraints*.

⁴Nebel and Bäckström [NB92] prove the NP-Completeness of a closely related case. Their proof technique is adapted here to prove Theorem 3.

Definition 11. (Regions) Given a set of events \mathcal{E} and a set of ordering constraints \mathcal{O} on \mathcal{E} , a subset X of \mathcal{E} forms a region if and only if the events in X occur as an atomic group while an event not in X occurring either before or after the events in X . The entire set of events \mathcal{E} and $\{e_i\}$ for every event e_i in \mathcal{E} are regions by themselves.

Example 2. In the bathroom example depicted in Figure 1, $X = \{e_1, e_2\}$, $Y = \{e_3, e_4\}$, $Z = \{e_5, e_6\}$, $W = X \cup Y \cup Z$ and every $\{e_i\}$ are regions.

Definition 12. (Child Regions) A region X is a child region of a region Y and Y is the parent region of X if and only if Y is the smallest region that contains X as a proper subset. A region X is a descendant region of a region Y and Y is an ancestor region of X if and only if X is a proper subset of Y .

Definition 13. (A Region Hierarchy) A region hierarchy is a set of regions in which for any two regions either they are disjoint or one is a descendant region of the other.

Example 3. In the bathroom example depicted in Figure 1, $X = \{e_1, e_2\}$, $Y = \{e_3, e_4\}$, and $Z = \{e_5, e_6\}$, are the child regions of $W = X \cup Y \cup Z$. $\{e_1\}$ and $\{e_2\}$ are the child regions of X . $\{e_3\}$ and $\{e_4\}$ are the child regions of Y . $\{e_5\}$ and $\{e_6\}$ are the child regions of Z . Together, they form a region hierarchy, which is depicted in Figure 1-c as a tree.

Definition 14. (Regional Ordering Constraints) The set of regional ordering constraints \mathcal{O}_R on a region R is a pair $\langle \sigma, \prec \rangle$, where $\sigma = \{R_1, R_2, \dots\}$ is the set of child regions of R , and \prec is a partial order on σ . The events in R_i must occur before the events in R_j if $R_i \prec R_j$.

Example 4. In the bathroom example depicted in Figure 1, the possible event sequences are determined by the region hierarchy and the following regional ordering constraints: $\mathcal{O}_X = \langle \{\{e_1\}, \{e_2\}\}, \emptyset \rangle$, $\mathcal{O}_Y = \langle \{\{e_3\}, \{e_4\}\}, \emptyset \rangle$, $\mathcal{O}_Z = \langle \{\{e_5\}, \{e_6\}\}, \{\{e_6\} \prec \{e_5\}\} \rangle$, and $\mathcal{O}_W = \langle \{X, Y, Z\}, \{Y \prec X\} \rangle$.

Remark. A region hierarchy and a set of regional ordering constraints describe a *non-interleaved hierarchical task network*. Tasks are organized as a hierarchy of regions. The task associated with a region R can be decomposed into the tasks associated with the child regions of R , and so on hierarchically. The tasks associated the child regions of a region R may be constrained by a partial order, but cannot be interleaved. A primitive task in the task network is an event, which is a disjunction of several STRIPS-like operators and it changes the system state by using exactly one of the applicable operators.

6. LOCALITY IN CAUSAL DEPENDENCIES

6.1. Causal Dependencies and Problem Decomposition

Events affecting all conditions are rare. Instead, events tend to affect or be affected by a subset of the set of conditions \mathcal{P} . If a condition p is irrelevant to the events in a region R , we do not need to directly concern ourselves with p in reasoning about the changes caused by the events in region R . This can result in significant computational savings.

Definition 15. (Causal Dependencies) A condition p is dependent on a causal rule $\alpha \rightarrow \beta$ if p appears in α or in β . A condition p is dependent on an event e if p is dependent on a causal rule associated with e . A condition p is dependent on a region R if p is dependent on an event in R .

Problem Decomposition: In the bathroom example (Figure 1-b and Figure 1-c), the conditions a , b , and c are dependent on region X , but independent of region Y ; the conditions d , e , and f are dependent on region Y , but independent of region X . If region Z were absent, we could decompose the problem instance into two independent parts: one with region X and the conditions a , b , and c ; the other one with region Y and the conditions d , e , and f . However, through the conditions a and d respectively, region Z interacts with the regions X and Y . This complicates the interactions among these three regions.

In general, two disjoint regions interact with one another through a few conditions that are dependent on both regions. In this situation, problem decomposition must be achieved by carefully exploiting structure in the causal dependencies among regions. In this section, we investigate the notions of the *subgoals* and *abstract events* of regions. These two notions reflect locality in causal dependencies. Using subgoals and abstract events, we are able to decompose the task of temporal projection into a sequence of local searches in individual regions.

6.2. Subgoals and Problem Decomposition

Given a region hierarchy, we can identify the *local conditions* and *subgoal conditions* of each individual region, which allows us to define the subgoals of regions.

Definition 16. (Local Conditions) A condition p is a local condition of region R if and only if p is dependent on R but not dependent on any events outside R .

Definition 17. (Subgoal Conditions) A condition p is a subgoal condition of a region R , if and only if (1) p is a local condition of region R , and (2) p is not a local condition of any child regions of R .

Example 5. Figure 4 displays the local conditions and subgoal conditions of the individual regions in the bathroom example.

The following lemmas directly come from the definitions of local conditions and subgoal conditions.

Lemma 2. If p is a local condition of a region R , the value of p can only be locally affected by the events in R . The value of p immediately before the events in R occur is the same as p 's initial value. Immediately after all of the events in R have occurred, the value of p will not be further changed.

Lemma 3. If p is a subgoal condition of a region R , then R is the smallest region containing p as a local condition. The sets of subgoal conditions of different regions are mutually disjoint and together they form a partition of the set of conditions \mathcal{P} .

Lemma 4. The set of local conditions of a region R is the union of the local conditions of R 's child regions and the subgoal conditions of R .

In the following, we define the regional subgoal and the incremental subgoal of a region.

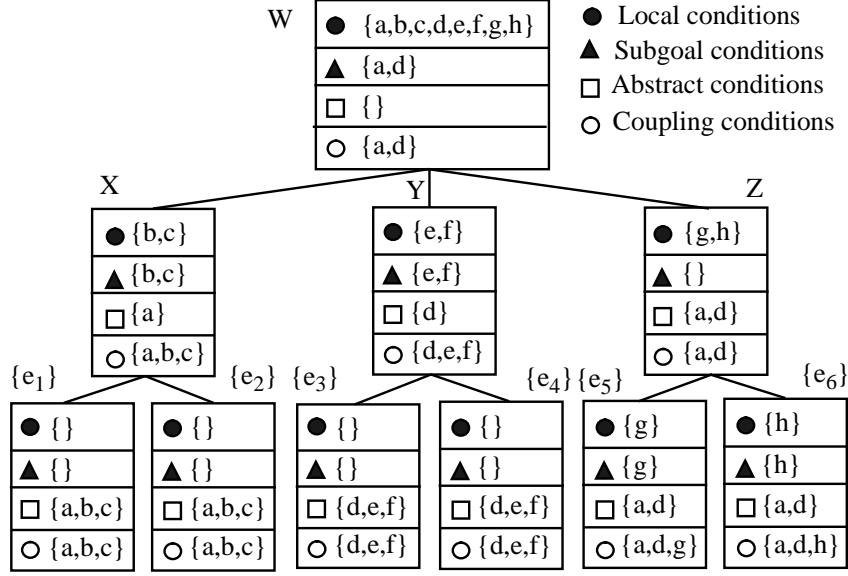


FIGURE 4. The condition subsets characterizing the causal dependencies among regions

Definition 18. (Regional Subgoals) Given a goal \mathcal{G} , the regional subgoal \mathcal{G}_R of a region R is composed of the condition/value pairs in \mathcal{G} whose condition components are local conditions of region R . The incremental subgoal $\tilde{\mathcal{G}}_R$ of region R is composed of the condition/value pairs in \mathcal{G} whose condition components are subgoal conditions of region R .

The following theorem indicates the relationship between the goal and the regional subgoals.

Theorem 7. The goal \mathcal{G} is true immediately after all events have occurred if and only if for each region R , the regional subgoal \mathcal{G}_R of R is true immediately after all of the events in R have occurred.

By Lemma 4 and the definition of regional subgoals and incremental subgoals, we have the following lemma.

Lemma 5. The regional subgoal \mathcal{G}_R of a region R is the union of the regional subgoals of R 's child regions and the incremental subgoal $\tilde{\mathcal{G}}_R$ of region R .

By Lemma 5, we have the following corollary of Theorem 7.

Corollary 2. In order to achieve goal \mathcal{G} , for each region R the regional subgoals of R 's child regions should be true respectively as soon as all of the events in the individual child regions have occurred; the incremental subgoal $\tilde{\mathcal{G}}_R$ should then be true immediate after all of the events in region R have occurred. Otherwise, the goal \mathcal{G} cannot be true after all events have occurred.

Example 6. Figure 5 depicts the incremental subgoals and the regional subgoals in the problem instance corresponding to the bathroom example. For example, the

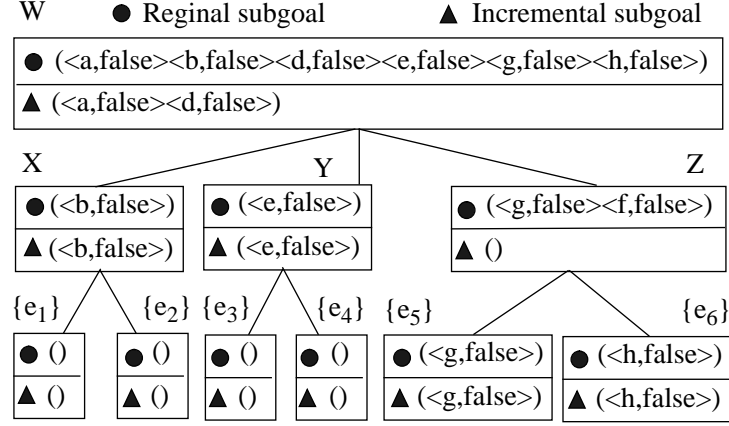


FIGURE 5. The regional subgoals and incremental subgoals of individual regions

regional subgoal of Y is $(\langle e, false \rangle)$, which indicates that condition e must be false immediately after all of the events in Y have occurred. The goal \mathcal{G} is equal to the regional subgoal \mathcal{G}_W of the root region W , which contains all events in the problem instance. To achieve the goal \mathcal{G} (i.e. \mathcal{G}_W), (1) the regional subgoals of regions X , Y , and Z must be achieved respectively as soon as the individual regions are finished, and then (2) the incremental subgoal for \mathcal{G}_W must be achieved immediately after all events have occurred.

6.3. The Abstract Event of a Region

Given a region hierarchy, we can identify the *abstract conditions* of each region, which allows us to define the abstract event of the corresponding region. Figure 4 displays the abstract conditions of individual regions in the bathroom example.

Definition 19. (Abstract Conditions) A condition p is an abstract condition of region R if p is (1) dependent on at least one event in R , and also (2) dependent on at least one event not in R .

Remark. Both the events in R and the events outside R can affect and be affected by the values of the abstract conditions of R . Therefore the set of abstract conditions of R is the media for inter-regional interactions between the events in R and the events outside R . For example, in Figure 1 the events e_1 and e_2 in region X interact with the events outside X through the abstract condition a of X .

Remark. The number of the abstract conditions of a region R is a measure of the locality regarding the causal dependencies between the events in R and the events outside R . If region R does not have any abstract conditions, all of the conditions involved in the events of R are local conditions of R . In this case, the events in R together with the local conditions of R can be separated from the other events and conditions to form an independent problem instance.

The effects of the events in a region R can only affect and be affected by the local conditions and the abstract conditions of R . Since the values of the local conditions of R immediately before the events in R occur are the same as their initial values, the state transitions triggered by the events in a region R are determined by two factors: (1) the values of the abstract conditions of R immediately before the events in R occur, and (2) the ordering of the events in R . To achieve the regional subgoals \mathcal{G}_R , (1) the set of abstract conditions of R must have appropriate values immediately before the events in R occur, and (2) the events in R must be properly ordered. In turn, these two factors also affect the values of the abstract conditions of R after achieving the regional subgoals \mathcal{G}_R . This information about the possible preconditions and their effects on the values of the abstract conditions can be compiled as an abstract event in achieving the regional subgoal. Before we define the abstract event of a region, we introduce the following terms.

Definition 20. (Abstract States) For an arbitrary subset $\mathcal{A} = \{p_1, \dots, p_h\}$ of the set of conditions \mathcal{P} , \mathcal{A} determines an abstract state space $S_{\mathcal{A}} = \{true, false\}^h$. The values of the conditions in \mathcal{A} at a specific point in time determines an abstract state in $S_{\mathcal{A}}$. An abstract state u of \mathcal{A} is represented as a vector $\langle v_1, \dots, v_h \rangle$ where $v_i, 1 \leq i \leq h$, indicates the value of condition p_i .

Definition 21. (Expressions and Abstract States) Let (1) \mathcal{A} and \mathcal{B} be two subsets of the set of conditions \mathcal{P} , $\mathcal{A} \subseteq \mathcal{B}$, and (2) u be an abstract state in $S_{\mathcal{B}}$. $\alpha_u^{\mathcal{A}}$ denotes the expression $\{\langle p_i, v_i \rangle | p_i \in \mathcal{A}, v_i \text{ is the value of } p_i \text{ in } u\}$.

$\alpha_u^{\mathcal{A}}$ is an expression that encodes the information about the values of the conditions in \mathcal{A} in an abstract state u . In the following, we formally define the abstract event of a region.

Definition 22. (Δ_R) Consider a region R where \mathcal{A} is the set of abstract conditions of R , and \mathcal{O} is the set ordering constraints. Δ_R is relation on $S_{\mathcal{A}} \times S_{\mathcal{A}}$ where $(u, v) \in \Delta_R$ if and only if there exists an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^R$ such that (1) u is the abstract state in $S_{\mathcal{A}}$ immediately before the events in R occur, (2) v is the abstract state in $S_{\mathcal{A}}$ immediately following \mathbf{q} , and (3) the regional subgoal \mathcal{G}_R is achieved in v .

Definition 23. (Abstract Events) An abstract event e_R of a region R is associated with a set of causal rules where $\alpha_u^{\mathcal{A}} \rightarrow \alpha_v^{\mathcal{A}}$ is an associated rule if and only if (u, v) is in Δ_R .

Remark. The relation Δ_R provides the information regarding the possible preconditions and their effects on the abstract conditions of R in order to achieve the regional subgoals \mathcal{G}_R . The abstract event e_R compactly represents Δ_R by abstracting away the information in Δ_R about the local conditions of region R . The abstract event e_R is what we need to reason about the inter-regional interactions between the events in R and the events outside R . An algorithm to derive abstract events is provided in Section 7.

Example 7. Figure 6 displays the abstract events of the regions regarding the bath-room example. At the top level, the global region W has no abstract conditions. Therefore $() \rightarrow ()$ is the only rule associated with the abstract event of the root region W . At the second level, it is not hard to see that a (d) must be true immediately

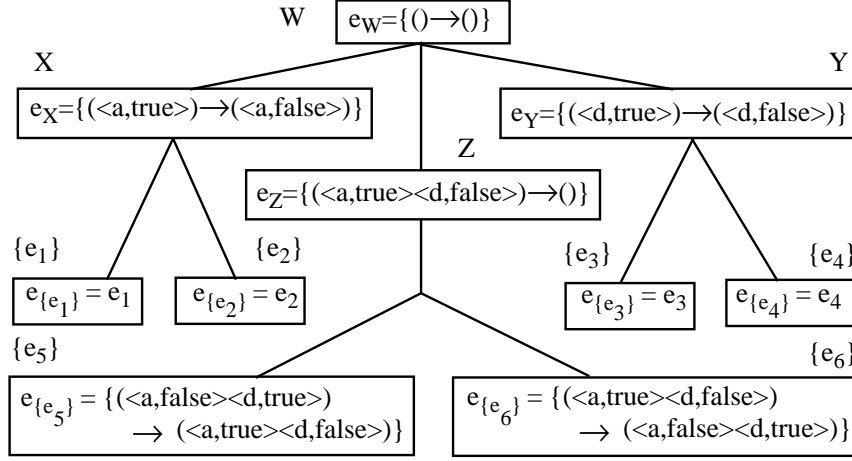


FIGURE 6. The abstract events of individual regions

before the events in region X (Y) to achieve the regional subgoal that b (e) is false, and consequently a (d) must be false immediately after region X (Y) is finished. This gives us the abstract events e_X and e_Y described in Figure 6. In region Z , (1) e_6 must occur before e_5 and (2) h (g) must be false immediately after e_6 (e_5) occurs. Therefore, a must be true and d must be false immediately before e_6 occur so that the first rule associated with e_6 can be applied to make h false. This also makes a false and d true immediately before e_5 occurs. When e_5 occurs following e_6 , it applies the first rule to make g false and also restores a to be true and d to be false. This gives us the abstract event $e_Z = \{(\langle a, true \rangle \langle d, false \rangle) \rightarrow ()\}$.

At the bottom level, except for regions $\{e_5\}$ and $\{e_6\}$ each region has an empty regional subgoal. In this situation, the possible preconditions and their effects to achieve the regional subgoal are directly inherent from the single event in the region. Therefore the abstract events for $\{e_1\}$, $\{e_2\}$, $\{e_3\}$ and $\{e_4\}$ are equal to e_1 , e_2 , e_3 and e_4 respectively.

For region $R = \{e_6\}$, the set of ordering constraints \mathcal{O} in R is empty. The only event sequence in $\mathbf{q} \in \mathbf{Q}_{\mathcal{O}}^R$ is $\langle e_6 \rangle$. $\mathcal{A} = \{a, d\}$ is the set of abstract conditions of R . Condition h is the local condition and also the subgoal condition of R . The regional subgoal \mathcal{G}_R is $\langle h, false \rangle$, and our task in region R is to make h false immediately after e_6 occurs. Note that h is true in the initial state. Therefore a must be true and d must be false immediately before e_6 occurs in order to apply the rule $(\langle a, true \rangle \langle d, false \rangle) \rightarrow (\langle a, false \rangle \langle d, true \rangle \langle h, false \rangle)$ to make h false. Consequently, a must become false and d must become true immediately after e_6 occurs. Therefore, immediately before the event sequence $\langle e_6 \rangle$ occurs the expression $\{\langle a, true \rangle \langle d, false \rangle\}$ must be true, while immediately after the event sequence $\langle e_6 \rangle$ occur the expression $\{\langle a, false \rangle \langle d, true \rangle\}$ must be true. This gives us the abstract event $e_{\{e_6\}} = \{(\langle a, true \rangle \langle d, false \rangle) \rightarrow (\langle a, false \rangle \langle d, true \rangle)\}$, which summarizes the underlying causal interactions as a causal rule. Similarly, we can derive the abstract event for region $\{e_5\}$ as described in Figure 6.

6.4. Coupling Conditions and Local Reasoning

In the following, we describe the *coupling conditions* of a region, which are the media for the inter-regional interactions among the child regions of R . Rather than considering the entire set of conditions \mathcal{P} , we only need to focus on the coupling conditions of a region R when reasoning about the events in R .

Definition 24. (Coupling Conditions) A condition p is a coupling condition of region R if and only if (1) region R is composed of two or more events and p is an abstract condition of a child region of R , or (2) region R is composed of a single event e and p is dependent on event e .

Example 8. Figure 4 depicts the coupling conditions of the individual regions in the bathroom example. For example, the three child regions X , Y , and Z of region W interact with one another through the coupling conditions a and d of region W .

Theorem 8. The set of coupling conditions of a region R is the union of two disjoint subsets: the set of abstract conditions of R and the set of subgoal conditions of R .

Remark. The child regions of a region R are coupled together through the coupling conditions of R . This is because by definition the union of the conditions appearing in these abstract events is exactly the set of coupling conditions of R . On the other hand, to derive an abstract event of region R , we need to reason about the values of the abstract conditions and the subgoal conditions of R , and their union is again the set of coupling conditions according to Theorem 8.

7. LOCALIZED TEMPORAL PROJECTION USING SUBGOALS AND ABSTRACT EVENTS

In this section, we reduce temporal projection into a sequence of local searches, in each of which we are concerned with the subgoal of a region R and the abstract events of the child regions of R .

7.1. Temporal Projection as Global Search

First of all, we describe how to formulate temporal projection as search. Given an instance of the temporal projection problem $I = \langle \mathcal{P}, \mathcal{E}, \mathcal{O}, s_0, \mathcal{G} \rangle$, the search graph is a directed graph $G_I = (V, A)$ that encodes the information of time, states, events in its vertices and arcs. Temporal projection is reduced to a graph reachability problem in the search graph.

Time and States: The set of vertices V represents the possible system states at different phases of the system's evolution. Each vertex contains two components: (1) a system state, and (2) a record indicating whether the individual events in \mathcal{E} have occurred or not.

Events: The set of arcs A represents the occurrences of events at different phases of the system's evolution. We construct an arc (u, v) in A if (1) by applying a causal rule r , an event e can cause a state transition from the state encoded in vertex u to

the state encoded in vertex v , (2) e can occur immediately after those events that are marked as having occurred in vertex u without violating the ordering constraints in \mathcal{O} , and (3) in vertex u , e is marked as having not occurred yet while in vertex v , e is marked as having occurred. In addition, we associate a *rule-tag* (e, r) with such an arc (u, v) to indicate that by applying rule r , event e can cause a state transition from the state encoded in vertex u to the state encoded in vertex v .

Root Vertex and Goal Vertices: The root vertex u_0 is the vertex in which no events have occurred yet, and the state in u_0 is the initial state s_0 . A vertex t is a goal vertex if all events have occurred in vertex t and the state in t is a goal state.

Temporal Projection as Search: The task of temporal projection can be viewed as a search for a path from the root vertex u_0 to any goal vertex t . Since an arc in $G_I = (V, A)$ models the occurrence of an event, such a path corresponds to a possible event sequence immediately following which we reach a goal state. The rule-tags on the arcs tell us the rules applied by the individual events to enter the goal state.

The following procedure **Temporal-Search** provides the details of temporal projection as search. Our localized algorithm utilizes this procedure for local search.

Procedure Temporal-Search

input: an instance of temporal projection problem $I = \langle \mathcal{P}, \mathcal{E}, \mathcal{O}, s_0, \mathcal{G} \rangle$.

output:

- (1) a set F composed of the goal states that can be entered immediately following some event sequences in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$;
- (2) an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ that ends in a goal state f in F ;
- (3) the rules applied by the individual events in \mathbf{q} to enter the goal state f ;
- (4) a trajectory $\mathbf{t}_{\mathbf{q}}$ of \mathbf{q} that ends in the goal state f .

1. Partition the set of events \mathcal{E} into m chains where the i th ($1 \leq i \leq m$) chain is composed of l_i events that are totally ordered as a chain according to the ordering constraints in \mathcal{O} . Construct a directed acyclic graph $G_I = (V, E)$ in step 2 and step 3.
2. Construct the vertex set V of G_I :
Each state s in $S_{\mathcal{P}}$ is mapped to $\prod_{1 \leq i \leq m} (1 + l_i)$ nodes, each of which indicates a specific phase in the system's evolution. A vertex t in V is of the form $(s, x_1, x_2, \dots, x_m)$. s ($s \in S_{\mathcal{P}}$) indicates the corresponding state in vertex t . x_i ($0 \leq x_i \leq l_i$, $1 \leq i \leq m$) is a counter for the i th chain; x_i indicates that the first x_i events in the i th chain have occurred while the remaining events in the i th chain have not occurred yet. These vertices comprise the vertex set V of G_I .
3. Construct the arc set A of G_I :
Add an arc from $(s, x_1, x_2, \dots, x_{k-1}, x_k - 1, x_{k+1}, \dots, x_m)$ to $(t, x_1, x_2, \dots, x_{k-1}, x_k, x_{k+1}, \dots, x_m)$ if (1) e is the x_k th event in the k th chain, (2) e can trigger a state transition from s to t by applying a causal rule r , and (3) e can occur immediately after the first $x_k - 1$ events in the k th chain and the first x_i events in the i th chain ($1 \leq i \leq m$, $i \neq k$) without violating the ordering constraints in \mathcal{O} . In addition, we associate a rule-tag (e, r) with this arc to indicate that event e can trigger such a state transition by applying causal rule r . These arcs comprise the arc set V of G_I , and model all possible situations regarding the occurrences of events.
4. In the following, we reduce the task of temporal projection to graph reachability.

We can apply any standard graph-reachability algorithms to derive a solution.

- Search the graph G_I . A goal state t can be reached immediately after an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ if and only if the goal vertex $(t, l_1, l_2, \dots, l_m)$ is reachable from the root vertex $(s, 0, 0, \dots, 0)$ in graph G_I . Add such a state t to the set F .
- Search for a directed path from the root vertex $(s, 0, 0, \dots, 0)$ to an arbitrary goal vertex $(f, l_1, l_2, \dots, l_m)$. The arcs in the directed path give us an event sequence \mathbf{q} that ends in the goal state f , and the rule-tags of the arcs tell us the rules applied by the individual events in \mathbf{q} to enter f . The state components of the vertices in the path give us a trajectory of \mathbf{q} which ends in f .

Theorem 9. The search graph $G_I = (V, E)$ constructed in procedure Temporal-Search is a directed acyclic graph. If we adopt a trivial partition that considers each individual event as a chain, the search graph G_I contains $2^{|\mathcal{P}|+|\mathcal{E}|}$ vertices. In this case, the procedure Temporal-Search runs in $O(2^{2^{(|\mathcal{P}|+|\mathcal{E}|)}})$ time.

7.2. Temporal Projection as a Sequence of Local Searches

In the following, we present a localized temporal projection algorithm. In this algorithm, two procedures are systematically called region by region: (1) procedure Abstract-Event, which derives an abstract event of a region, and (2) procedure Generate-Sequence, which recursively generates an event sequence to achieve the regional subgoal of a region. Both procedures in turn call procedure Temporal-Search to conduct local search. For a region R , the following causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \mathcal{G}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ is provided as input to procedure Abstract-Event and procedure Generate-Sequence, where

- \mathcal{A}_R is the set of abstract conditions of R ;
- \mathcal{U}_R is the set of subgoal conditions of R ;
- $\mathcal{C}_R = \mathcal{A}_R \cup \mathcal{U}_R$ is the set of coupling conditions of R ;
- s_0 is the initial state of the problem instance;
- \mathcal{G}_R is the incremental subgoal of R ;
- if R contains more than one event, \mathcal{X}_R is the set of abstract events \mathcal{X}_R of R 's child regions; otherwise \mathcal{X}_R is composed of the single event in region R ;
- \mathcal{O}_R is the set of regional ordering constraints on the child regions of R .

Procedure Localized-Reasoning

input: a problem instance $\langle \mathcal{P}, \mathcal{E}, \mathcal{O}, s_0, \mathcal{G} \rangle$.

output: Report an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ and the rules applied by the events in \mathbf{q} such that we enter a goal state immediately following \mathbf{q} if such an event sequence exists; otherwise, report that such a sequence does not exist.

1. Establish the following causal knowledge by scanning through the problem instance: (i) the regions, the region hierarchy, and the regional ordering constraints, (ii) the local conditions, the subgoal conditions, the abstract conditions and the coupling conditions of individual regions, (iii) the regional subgoals and incremental subgoals of individual regions.

2. Starting from the bottom level of the region hierarchy, call procedure Abstract-Event as described in Step 3 for each region at the same level; proceed in the same way level by level until the root region at the top level is done.
3. For each region R , call procedure Abstract-Event to determine the possibility of achieving the regional subgoal of R . If the regional subgoal cannot be achieved, abort the computation and report the infeasibility of achieving the goal \mathcal{G} ; otherwise, derive and propagate the abstract event e_R to R 's parent region.
4. Call procedure Generate-Sequence with a pair of empty abstract states $(\langle \rangle, \langle \rangle)$ and the causal knowledge $\langle \mathcal{A}_{\mathcal{E}}, \mathcal{U}_{\mathcal{E}}, \mathcal{C}_{\mathcal{E}}, s_0, \tilde{\mathcal{G}}_{\mathcal{E}}, \mathcal{X}_{\mathcal{E}}, \mathcal{O}_{\mathcal{E}} \rangle$ associated with the root region \mathcal{E} as input. Procedure Generate-Sequence recursively generates an event sequence \mathbf{q} and the rules applied by the events in \mathbf{q} to achieve the goal \mathcal{G} .

Procedure Abstract-Event

Input: the causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ associated with region R .
Output: if the incremental subgoal of R can be achieved, report the abstract event e_R ; otherwise, stop and report the infeasibility of achieving the regional subgoal.

1. Determine the set S of the possible abstract states in $S_{\mathcal{C}_R}$ immediately before the events in R occur, where the values of the subgoal conditions in \mathcal{U}_R must be the same as their initial values in s_0 .
2. For each abstract state s in S , construct a new instance of the temporal projection problem $\langle \mathcal{C}_R, \mathcal{X}_R, \mathcal{O}_R, s, \tilde{\mathcal{G}}_R \rangle$ regarding the set of coupling conditions \mathcal{C}_R , the set of abstract events \mathcal{X}_R , the set of regional ordering constraints \mathcal{O}_R , the possible abstract state s immediately before the events in R occur, and the incremental subgoal $\tilde{\mathcal{G}}_R$.
3. For each new instance $\langle \mathcal{C}_R, \mathcal{X}_R, \mathcal{O}_R, s, \tilde{\mathcal{G}}_R \rangle$, call procedure Temporal-Search to derive F , the set of reachable goal states in $S_{\mathcal{C}_R}$ regarding this new instance. Add the pairs (s, f) for each f in F into Δ_R .
4. If the set Δ_R is empty, we report the infeasibility of achieving the incremental subgoal. Otherwise, compile Δ_R into an abstract event e_R by associating a causal rule $\alpha_s^{\mathcal{A}_R} \rightarrow \alpha_f^{\mathcal{A}_R}$ with e_R for each pair of abstract states (s, f) in Δ_R .

Procedure Generate-Sequence

Input:

- (1) the causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ associated with region R ;
- (2) a pair of abstract states (s', f') in $S_{\mathcal{A}_R}$ where s' and f' are the abstract states immediately before and after the events in R occur as a group respectively.

Output: an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^R$ and the rules applied by the events such that if the abstract state s' is true immediately before \mathbf{q} , immediately following \mathbf{q} the abstract state is changed to f' and the regional subgoal \mathcal{G}_R is true.

1. Determine the abstract states s in $S_{\mathcal{C}_R}$ where (i) the values of the conditions in \mathcal{A}_R in s are the same as those in abstract state s' , and (ii) the values of the conditions in \mathcal{U}_R in s are the same as their initial values in s_0 .

2. Construct a new instance of the temporal projection problem $\langle \mathcal{C}_R, \mathcal{X}_R, \mathcal{O}_R, s, \tilde{\mathcal{G}}_R \rangle$ regarding the set of coupling conditions \mathcal{C}_R , the set of abstract events \mathcal{X}_R , the set of regional ordering constraints \mathcal{O}_R , the possible abstract state s immediately before the events in R occur, and the incremental subgoal $\tilde{\mathcal{G}}_R$.
3. For the new instance $\langle \mathcal{C}_R, \mathcal{X}_R, \mathcal{O}_R, s, \tilde{\mathcal{G}}_R \rangle$, call procedure Temporal-Search to derive an event sequence \mathbf{q}' in $\mathbf{Q}_{\mathcal{O}_R}^{\mathcal{X}_R}$ that ends in a reachable goal state f where (i) the incremental subgoal $\tilde{\mathcal{G}}_R$ is true in f , and (ii) the expression $\alpha_f^{\mathcal{A}_R}$ is true in f .
4. Also use procedure Temporal-Search to determine (i) a trajectory $\mathbf{t}_{\mathbf{q}'}$ of \mathbf{q}' that starts in s and ends in f , and (ii) the rules applied by the events in \mathbf{q}' that results in the trajectory $\mathbf{t}_{\mathbf{q}'}$. The ordering of the abstract events in \mathbf{q}' determines a total order on the child regions of R .
5. For each child region R' of R , according to the trajectory $\mathbf{t}_{\mathbf{q}'}$ of \mathbf{q}' derived in Step 3, determine the abstract states \tilde{s} and \tilde{f} in $S_{\mathcal{A}_{R'}}$ immediately before and after the abstract event of R' in \mathbf{q}' occurs respectively.
6. For each child region R' of R , (i) prepare the pair of abstract states (\tilde{s}, \tilde{f}) derived in Step 5 and the causal knowledge $\langle \mathcal{A}_{R'}, \mathcal{U}_{R'}, \mathcal{C}_{R'}, s_0, \tilde{\mathcal{G}}_{R'}, \mathcal{X}_{R'}, \mathcal{O}_{R'} \rangle$ associated with R' as input, and (ii) call procedure Generate-Sequence to derive an event sequence over R' . Concatenate these event sequences over the child regions of R into an event sequence \mathbf{q} according to the total order on R 's child regions derived in Step 4.
7. For each event e , determine the rule applied by event e according to the rule selected in Step 4 when procedure Generate-Sequence is recursively called to handle the region $\{e\}$.

Theorem 10. Procedure Localized-Reasoning is sound and complete.

Example 9. We illustrate the use of the localized algorithm in solving our bathroom example. First, we derive the following knowledge by scanning through the problem instance: (1) the region hierarchy as depicted in Figure 4; (2) the local conditions, the subgoal conditions, the abstract conditions and the coupling conditions of the individual regions as depicted in Figure 4; and (3) the regional subgoals and incremental subgoals of individual regions as depicted Figure 5.

Second, starting from the bottom level of the region hierarchy, we call procedure Abstract-Event to derive the abstract events of regions as depicted in Figure 6. We proceed level by level until we reach the root region W at the top level. Since we can successfully derive the abstract event for every region, there exists an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ to achieve the goal.

Finally, we call procedure Generate-Sequence to recursively derive an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ to achieve the goal. The procedure Generate-Sequence recursively generates the following orderings $\langle Y, Z, X \rangle$, $\langle \{e_3\}, \{e_4\} \rangle$, $\langle \{e_6\}, \{e_5\} \rangle$, and $\langle \{e_1\}, \{e_2\} \rangle$ for the child regions of the regions W , Y , Z , and X respectively. This gives us an event sequence $\mathbf{q} = \langle e_3, e_4, e_6, e_5, e_1, e_2 \rangle$ as an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ to achieve the goal. The causal rules applied by the events in \mathbf{q} as a sequence are the first rule, the second rule, the first rule, the first rule, the first rule, and the second rule of the events respectively.

8. QUANTIFYING THE COMPUTATIONAL EFFICIENCY

We first define the *interaction measure* of a problem instance, which is used to quantify the computational efficiency of procedure Localized-Reasoning in the problem instance.

Definition 25. (Measure of Causal Interactions) Given the region hierarchy determined by an instance of the temporal projection problem, we define the branching factor b_R of a region R to be the number of child regions of R , the coupling factor c_R of a region R to be the number of coupling conditions of R , and the interaction measure \mathcal{L} of the problem instance to be $\max_R (b_R + c_R)$.

The interaction measure \mathcal{L} indicates the magnitude of the interactions among regions. A large interaction measure \mathcal{L} indicates the existence of regions with many child regions or large sets of coupling conditions, which implies complicated interactions among regions. Instead, a small interaction measure \mathcal{L} indicates that the causal interactions are well structured and localized in individual regions. In other words, *small* interaction measure \mathcal{L} implies *strong* locality in the problem instance.

Theorem 11. Given an instance of the temporal projection problem, procedure Localized-Reasoning runs in $O(n \times 8^{\mathcal{L}})$ time, where n is the size of the event set \mathcal{E} , and \mathcal{L} is the interaction measure of the problem instance.

By Theorem 11, the worst-case performance degrades to be exponential in n when the branching factor or the coupling factor is of $O(n)$ size, where n is the size of the event set \mathcal{E} . This is as expected, because (1) a set of totally unordered events \mathcal{E} corresponds to a single region containing the individual events as child regions, where the branching factor is of $O(n)$ size, and (2) temporal projection regarding totally unordered events is NP-complete even in very restricted cases as shown in Theorem 2.

On the other hand, our localized algorithm can exploit the locality inherent in problem instances to bring about computational efficiency. When there is strong locality in a problem instance, the interaction measure \mathcal{L} is small. In this situation, our localized reasoning algorithm can substantially expedite temporal projection. The following two corollaries of Theorem 11 demonstrate the computational savings gained by exploiting locality.

Corollary 3. When the interaction measure \mathcal{L} is of $O(1)$ magnitude, the temporal projection problem can be solved by procedure Localized-Reasoning in $O(n)$ time, where n is the size of the event set \mathcal{E} .

Corollary 4. When the interaction measure is of $O(\log n)$ magnitude, the temporal projection problem can be solved by procedure Localized-Reasoning in time polynomial in n , where n is the size of the event set \mathcal{E} .

Remark. Consider a situation in which (1) every region except for the regions containing only a single event has \tilde{b} child regions, and (2) for each region the child regions of the region are totally unordered.

- There are $\tilde{b}!^{(n-1)/(\tilde{b}-1)}$ possible total orders on events, where n is the size of the event set \mathcal{E} . It is infeasible to exhaustively examine this super-polynomial number of possible total orders one by one for a solution.

- However, procedure Localized-Reasoning can derive a solution in linear time if (1) $\tilde{b} = O(1)$ and (2) every region has only $O(1)$ number of conditions that are dependent on both the events in and the events not in the region. The interaction measure \mathcal{L} in this case is of $O(1)$ magnitude. This is because for each region R (1) $b_R = \tilde{b} = O(1)$ and (2) the coupling factor c_R is no more than the sum over the $O(1)$ numbers of the abstract conditions of R 's child regions according to Theorem 8.
- Similarly, procedure Localized-Reasoning can derive a solution in polynomial time if (1) $\tilde{b} = O(\log n)$ and (2) every region has only $O(\log n)$ number of conditions that are dependent on both the events in and the events not in the region. The interaction measure \mathcal{L} in this case is of $O(\log n)$ magnitude. In both cases, procedure Localized-Reasoning provides substantial computational savings.

9. RELATED WORK

The related work on the representation of time and temporal reasoning is extensive. There are several formalisms that can represent the temporal relationship among a set of points or intervals in time [All83] [Vi82] [MB83] [Dea84]. Reasoning about the consequence of events using the logic-based event calculus can be found in [KS86] [Chi94]. The idea of structuring ordering and causal knowledge to speed up temporal reasoning is also present in maintaining knowledge about temporal intervals [All83] [GS93] [KG77] and in the event calculus [Eva90] [Mea92].

The use of subgoals and abstract events in this paper is inspired by the similar notions of subgoals and abstraction used in search [Kor87] and in abstract and hierarchical planning [Sac74] [Chr90] [Kno91] [YT90] in reducing the overall search effort. The idea of using localized reasoning to exploit locality is also studied by Lansky in GEM [Lan88] in event-based planning. In GEM, locality is similarly modeled by sets of interrelated events delimited by temporal logic constraints.

10. CONCLUSION

In this paper, we formally describe a dynamical system modeled by a set of conditions, a set of events, and a set of ordering constraints on the events. The system state at a point in time is determined by the values of a finite set of conditions. We represent an event as a disjunction of STRIPS-like operators. The system evolves through a sequence of state transitions as the events occur and change the values of conditions. We are concerned with deriving an event sequence consistent with the ordering constraints to achieve a goal. We identify the factors affecting computational complexity and demonstrate the complexity trade-offs involving these factors. Our complexity results indicate that we have very limited ability to consider a large number of conditions at a time. To expedite temporal projection, we investigate locality in event ordering and causal dependency. Locality allows us to focus on many smaller numbers of conditions separately rather than considering the entire set of conditions at a time. The locality in event ordering in a particular problem instance is modeled by a hierarchy of regions. A region is a collection of closely related events which occur as an atomic group. For each region, we can identify four characteristic condition subsets, which reflect locality in causal dependencies. We

develop a localized temporal reasoning algorithm that determines a solution through a sequence of local searches. The computational complexity of our algorithm is determined by the locality of a problem instance. Our theoretical results demonstrate the substantial performance improvement gained by exploiting locality. This work provides a solid evidence of the usefulness of localized reasoning in exploiting locality.

A. PROOFS

Theorem 1. Given an instance of the temporal projection problem, the PRJ task, the PRJ1 task, and the PRJ2 task can be reduced to one another by transforming the problem instance. All these transformations can be done in polynomial time by (1) adding at most two more conditions and two more events, and (2) appropriately modifying the ordering constraints and causal rules.

Proof. Given an instance I for the PRJ task, we can construct the following instance I' for the PRJ2 task. We add a new condition p and a new event e into I such that the only effect of event e is to make p true unconditionally. Condition p is constrained to be false initially. Event e is constrained to happen after the other events. We add $\langle p, true \rangle$ into the goal. It is obvious that the goal in instance I can be true immediately after all events have occurred if and only if the new goal in instance I' can be true immediately after event e has occurred.

Given an instance I for the PRJ2 task, we can construct the following instance I' for the PRJ1 task. Suppose e is the specified tail event in the PRJ2 task. We add a new condition p and add a new event e' . Event e' makes p true if the goal in instance I is true immediate before e' occurs. Condition p is constrained to be false initially. Event e' is constrained to happen immediately following e . $\langle p, true \rangle$ is the new goal. It is obvious that the goal of instance I can be true immediately after the specified event e has occurred if and only if the new goal can be true immediately following a possible event sequence in the new instance I' .

Given an instance I for the PRJ1 task, we can construct the following instance I' for the PRJ task. We add a new condition p . All events remain the same except that in addition every event makes p true if the goal in instance I is true immediate before the event occurs. Condition p is constrained to be false initially. $\langle p, true \rangle$ is the new goal. It is obvious that the goal of instance I can be true immediately following a possible event sequence if and only if the new goal in instance I' can be true after all events have occurred.

Each of the transformations above adds at most one more condition and one more event. Since the tasks can be reduced to one another by composing at most two of the transformations above, at most two more conditions and two more events will be added into the final transformed problem instance. ■

Theorem 2. Temporal projection regarding two-rule events, totally unordered, with a polynomial-size state space is NP-Complete.

Proof. Given a directed graph with a set of forbidden pairs of arcs, the forbidden pairs of arcs problem [GJ79] determines whether there is a path between two nodes without using both arcs in any forbidden pair of arcs. It is NP-Complete even if all forbidden pairs are disjoint.

In the following, we reduce an instance of the disjoint forbidden pairs of arcs problem to an instance of the temporal projection problem regarding two-rule events, totally unordered, with a polynomial-size state space. (1) Each non-isolated vertex in the directed graph is mapped to a distinct state in a state space. In particular, the starting vertex s and the target vertex t are mapped to the initial state and a unique goal state respectively. (2) Each directed arc (u, v) in the graph is mapped to a distinct causal rule that is only applicable at state u and causes a state transition from u to v . (3) Each forbidden pair of arcs is mapped to a distinct two-rule event which is associated with the two causal rules that correspond to the two arcs in the forbidden pair. Events are allowed to occur in arbitrary order.

Since an event corresponds to a forbidden pair of arcs, the event can trigger a state transition along exactly one of the two arcs. Therefore an event sequence with a trajectory starting from s and ending in t corresponds to a solution directed path in the forbidden pairs of arc problem instance. On the other hand, each arc in the directed graph is uniquely associated with a two-rule event. Therefore a directed path from s to t in the directed graph corresponds to an event sequence starting from s and ending in t . This reduction together with Lemma 1 and Theorem 1 prove the NP-Completeness of this special case of temporal projection. ■

Theorem 3. Temporal projection regarding one-rule events with an arbitrary partial order and a polynomial-size state space is NP-Complete.

Proof. Given a directed graph with a set of forbidden pairs of arcs, the forbidden pairs of arcs problem [GJ79] determines whether there is a path between two nodes without using both arcs in any forbidden pair of arcs. This problem is NP-Complete even for directed acyclic graphs.

In the following, we reduce an instance of the forbidden pairs of arcs problem on a directed acyclic graph to an instance of the temporal projection problem regarding one-rule events with an arbitrary partial order and a polynomial-size state space. (1) Each non-isolated vertex in the directed graph is mapped to a distinct state in a state space. In particular, the starting vertex s and the target vertex t are mapped to the initial state and a unique goal state respectively. (2) Each directed arc (u, v) in the graph is mapped to a distinct causal rule that is only applicable at state u and causes a state transition from u to v . Each causal rule is then uniquely associated with a one-rule event. In other words, each arc is uniquely mapped to a one-rule event. (3) For each forbidden pair of arcs (e, e') where e and e' correspond to the one-rule events e_1 and e_2 respectively, we impose an ordering constraint $e_1 \prec e_2$ ($e_2 \prec e_1$) if there is a directed path with arc e' preceding arc e (e preceding arc e'). Since the graph is acyclic, these ordering constraints define a partial order on the events.

For a forbidden pair of arcs (e, e') where e and e' correspond to the one-rule events e_1 and e_2 respectively, the imposed ordering constraint on the events e_1 and e_2 prevents the arcs e and e' from both appearing in the same trajectory of any possible event sequence. Note that each arc is uniquely mapped to a one-rule event. Given the ordering constraints, event e_1 must precede event e_2 if there is a directed path in the graph with arc e' preceding arc e . However, event e_1 precedes event e_2 implies that a transition along the arc e must precede a transition along arc e' if both transitions are contained in the trajectory of an event sequence. However, this is impossible, since in a directed acyclic graph the existence of a directed path with arc e' preceding arc e implies the nonexistence of a path with arc e preceding arc e' . This proves that arcs e and e' can not both appear in the trajectory of any possible

event sequence if (e, e') is a forbidden pair of arcs. Note that each arc is uniquely mapped to a one-rule event. Therefore a possible event sequence with a trajectory starting from s and ending in t gives us a solution directed path in the forbidden pairs of arc problem instance and vice versa. This reduction together with Lemma 1 and Theorem 1 prove the NP-Completeness of this special case of temporal projection. ■

Theorem 4. Temporal projection regarding one-rule events, totally unordered, with a polynomial-size state space is solvable in polynomial time.

Proof. For this special case, the PRJ1 task of temporal projection can be reduced to a graph reachability problem, which is solvable in polynomial time [CLR91]. We first construct a directed graph G by mapping (1) each state to a distinct vertex in G and (2) each one-rule event that triggers a state transition from a state u to a state v to an arc (u, v) in G . Finding an event sequence that ends in a goal state is equivalent to finding a path from s to t in graph G where s and t correspond to the initial state and a goal state respectively. This reduction together with Theorem 1 prove that this special case of temporal projection is solvable in polynomial time. ■

Theorem 5. Temporal projection regarding general events partially ordered as $O(1)$ number of chains of length $O(|\mathcal{E}|)$ or $O(\log |\mathcal{E}|)$ number of chains of $O(1)$ length with a polynomial-size state space can be solved in polynomial time.

Proof. Given a problem instance $\langle \mathcal{P}, \mathcal{E}, \mathcal{O}, s_0, \mathcal{G} \rangle$, the number of vertices in the graph constructed by the global search algorithm for temporal projection in Section 7 is $|\mathcal{S}_{\mathcal{P}}| \times \prod_{1 \leq i \leq m} (1 + l_i)$, where $|\mathcal{S}_{\mathcal{P}}|$ is the size of the state space, m is the number of chains, and l_i is the number of events in the i th chain. If $l_i = O(|\mathcal{E}|)$ and m is a constant, or $l_i = O(1)$ and $m = O(\log |\mathcal{E}|)$, in both cases the graph G is polynomial in the size of the event set \mathcal{E} . Therefore graph reachability and thus temporal projection is solvable in polynomial time for this special case of temporal projection. ■

Theorem 6. Temporal projection regarding general events with an arbitrary partial order and a constant-size state space is solvable in polynomial time.

Proof. For this special case, the task in the PRJ1 task of temporal projection can be accomplished in the following way. First, we construct a directed graph G by (1) mapping each state to a distinct vertex in G , and (2) adding an arc (u, v) to G if one of the events can trigger a state transition from u to v . G is a graph of $O(1)$ size.

Second, we enumerate all simple directed paths which start from the initial state and end in goal states. There is only a constant number of such directed paths since the graph is of constant size. For each of these directed paths, we must determine whether there is an event sequence whose trajectory can correspond to the directed path. This task is equivalent to bipartite matching between the directed arcs in a simple directed path and the set of events. An event e can match a directed arc (u, v) if e can trigger a state transition from u to v . If each of the directed arcs in the directed path can be matched with a distinct event, this gives us an event sequence whose trajectory is the directed path. Finally, we examine the event sequence to determine whether it is consistent with the ordering constraints. Since the bipartite match problem can be solved in polynomial time [PS82] [CLR91], this reduction together with Theorem 1 prove that this special case of temporal projection is solvable in polynomial time. ■

Theorem 7. The goal \mathcal{G} is true immediately after all events have occurred if and only if for each region R , the regional subgoal \mathcal{G}_R of R is true immediately after all of the events in R have occurred.

Proof. According to the definition, all regional subgoals are subsets of the goal \mathcal{G} . In particular, the goal \mathcal{G} equals the regional subgoal of the root region, since every condition is a local condition in this global region. Therefore the goal \mathcal{G} is true after all events occur if and only if for each region R , the regional subgoal \mathcal{G}_R is true after all events occur. By Lemma 2, a local condition p of a region R is true after all events occur if and only if p is true immediately after the events in region R have all occurred. Therefore a regional subgoal \mathcal{G}_R is true after all events occur if and only if \mathcal{G}_R is true immediately after the events in region R have all occurred. ■

Theorem 8. The set of coupling conditions of a region R is the union of two disjoint subsets: the set of abstract conditions of R and the set of subgoal conditions of R .

Proof. The set of abstract conditions of R and the set of subgoal conditions of R are disjoint, since the abstract conditions are dependent on events not in R and the subgoal conditions of R are local conditions of R . In the following, we consider a region R of two or more child regions. (For a region composed of a single event, the proof directly follows from the definitions of these three sets of conditions.) Suppose p is an abstract condition of R . p is dependent on an event not in R and an event e in R . Since e is contained in one of R 's child regions R' , p is also an abstract condition of this child region R' . If p is a subgoal condition of R , then p must be an abstract condition of one of R 's child regions. This is because p is dependent on R , but not a local condition of any child region of R . Therefore, if p is in the union of the set of abstract conditions and the set of coupling conditions of R , then p must be an abstract condition of a child region R' of R , which implies that p is also a coupling condition of R .

On the other hand, suppose p is a coupling condition of R . p must be an abstract condition of a child region R' of R . If p is not an abstract condition of R , p is a local condition of R which is only dependent on the events in R . Since p is an abstract condition of a child region R' of R , p must be dependent on an event in R but not in R' . This implies that p is not a local condition of any child region of R . In other words, p is a subgoal condition of R . Therefore if p is a coupling condition of R , then p is either an abstract condition of R or a subgoal condition of R . ■

Theorem 9. The search graph $G_I = (V, E)$ constructed in procedure Temporal-Search is a directed acyclic graph. If we adopt a trivial partition that considers each individual event as a chain, the search graph G_I contains $2^{|\mathcal{P}|+|\mathcal{E}|}$ vertices. In this case, the procedure Temporal-Search runs in $O(2^{2^{(|\mathcal{P}|+|\mathcal{E}|)}})$ time.

Proof. When we follow a directed arc (u, v) in E , exactly one of the time counters is increased by one in v compared with the counters in u , and the other time counters remain the same. Consider the counters associated with the individual chains. There is no directed path from u to u where the counters are the same in the head and the tail, since at least one of the counters is increased by following a directed path. Therefore $G_I = (V, E)$ is a directed acyclic graph, and the single-source shortest paths problem and thus the graph reachability problem can be solved in $O(|V| + |E|)$

time [CLR91]. If we adopt the trivial partition, we have $|\mathcal{E}|$ chains, and the length l_i equals one for each chain. This gives us $|\mathcal{S}_{\mathcal{P}}| \times \prod_{1 \leq i \leq m} (1 + l_i) = 2^{|\mathcal{P}| + |\mathcal{E}|}$ vertices, and at most $2^{2(|\mathcal{P}| + |\mathcal{E}|)}$ arcs. Therefore the procedure Temporal-Search runs in $O(2^{2(|\mathcal{P}| + |\mathcal{E}|)})$ time. \blacksquare

Theorem 10. Procedure Localized-Reasoning is sound and complete.

Proof. By induction on the levels of the region hierarchy, we can establish the following two properties about the localized algorithm. First, for each region R , the regional subgoal \mathcal{G}_R can be true immediately after the events in R have all occurred if and only if (1) procedure Abstract-Event can successfully derive the abstract event e_R and (2) at least one of the rules associated with e_R is applicable immediately before the events in R occur. Second, for each region R , procedure Generate-Sequence can generate an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^R$ immediately following which \mathcal{G}_R can be true if and only if (1) s' is the abstract state in $\mathcal{S}_{\mathcal{A}_R}$ immediately before the events in R occur where $\langle s', t' \rangle$ is the pair of abstract states in $\mathcal{S}_{\mathcal{A}_R}$ given as the input to procedure Generate-Sequence, and (2) at least one of the rules associated with e_R is applicable in s' .

The induction hypothesis is composed of the two properties described above. In the basis step, we consider every region that is composed of a single event. Both properties are true in this situation directly following the localized algorithm and the definition of abstract events. In the induction step, we consider every region R in which both properties in the induction hypothesis are true for every child region of R . According to procedure Abstract-Event and the definitions of Δ_R and e_R , for each rule $\alpha \rightarrow \beta$ associated with e_R , if α is true immediately before the events in R occur, we have the following two implications: (a) the abstract events of R 's child regions can occur as a sequence such that for each child region R' , at least one of the rules associated with the abstract event $e_{R'}$ is applicable immediately before $e_{R'}$ occur, and (b) the incremental subgoal $\tilde{\mathcal{G}}_R$ is true immediately after the events in R occur. These two implications together with Corollary 2 allows us to complete the induction step for the two proposed properties.

First, suppose that we can derive the abstract event e_R for R . By implication (a) and the induction hypothesis, for each child region R' of R , the regional subgoal $\mathcal{G}_{R'}$ can be true immediately after the events in R' have all occurred. Therefore, according to implication (b) and Corollary 2, the regional subgoal \mathcal{G}_R can be true immediately after the events in R have all occurred if one of the rules associated with e_R is applicable immediately before the events in R occur. On the other hand, if we can not derive the abstract event e_R , this implies that Δ_R is empty and the regional subgoal \mathcal{G}_R can never be achieved. This completes the induction step for the first property.

Second, suppose that (1) for a region R , $\langle s', t' \rangle$ is the pair of abstract states in $\mathcal{S}_{\mathcal{A}_R}$ given as the input to procedure Generate-Sequence and (2) at least one of the rules associated with e_R is applicable in s' . By the implications (a) and (b) and the induction hypothesis, if one of the rules associated with e_R is applicable immediately before the events in R occur, we are able to generate an event sequence \mathbf{q} in $\mathbf{Q}_{\mathcal{O}}^R$ such that (1) for each child region R' of R , the regional subgoal $\mathcal{G}_{R'}$ is true immediately after the events in R' have all occurred and (2) the incremental subgoal $\tilde{\mathcal{G}}_R$ is true immediately following \mathbf{q} . This completes the induction step for the first property.

By induction, both properties are true for all regions. In particular, consider the root region \mathcal{E} . Note that (1) since all conditions are local conditions in the root

region, the root region has an empty set of abstract conditions; (2) $() \rightarrow ()$ is the only rule in the abstract event of the root region if we can successfully derive the abstract event; and (3) in particular, in Step 4 of procedure Localized-Reasoning, $()$ is true in $s' = \langle \rangle$ immediately before the events in \mathcal{E} occur where $(s', f') = (\langle \rangle, \langle \rangle)$ is the pair of abstract states given to procedure Generate-Sequence. Therefore, according to the two properties, procedure Localized-Reasoning always successfully (1) determines the existence of an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ that ends in a goal state and (2) generates such an event sequence if and only if one exists. ■

Theorem 11. Given an instance of the temporal projection problem, the procedure Localized-Reasoning runs in $O(n \times 8^{\mathcal{L}})$ time, where n is the size of the event set \mathcal{E} , and \mathcal{L} is the interaction measure of the problem instance.

Proof. In the procedure Localized-Reasoning, the procedure Region-Abstraction and the procedure Generate-Sequence are (recursively) called once in each region, and both procedures use the procedure Temporal-Search for local searches in individual regions. In particular, in each region the procedure Temporal-Search is called $2^{|\mathcal{A}_R|}$ times in Step 2 of the procedure Region-Abstraction. In a region R , procedure Temporal-Search runs in $O(2^{2(b_R+c_R)})$ time according to Theorem 9, since there are b_R child regions and c_R coupling conditions in R . Therefore procedure Localized-Reasoning runs in $\sum_R 2^{|\mathcal{A}_R|} \times O(2^{2(b_R+c_R)})$ time, which can be written as $O((2n-1) \times 2^{\mathcal{L}} \times 2^{2\mathcal{L}})$ according to the following two observations: First, in a region hierarchy, the number of distinct regions is no more than $2n-1$; this is because in a tree with n leaves, there is at most $n-1$ internal vertices, which corresponds to a region hierarchy with n events. Second, in each region, we have

$$\mathcal{L} \geq b_R + c_R \geq c_R = |\mathcal{C}_R| = |\mathcal{U}_R \cup \mathcal{A}_R| \geq |\mathcal{A}_R|.$$

Accordingly we have a $O(n \times 8^{\mathcal{L}})$ time bound. ■

REFERENCES

- [All83] James Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM*, 26:832–843, 1983.
- [BK91] C. Bäckström and I. Klein. Parallel non-binary planning in polynomial time. In *Proceedings IJCAI 12*, pages 268–273. IJCAI, 1991.
- [Byl91] Tom Bylander. Complexity results for planning. In *Proceedings IJCAI 12*, pages 274–279. IJCAI, 1991.
- [Cha87] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–377, 1987.
- [Chi94] L. Chittaro. Skeptical and credulous event calculi for supporting modal queries. In *Proceedings ECAI-94*, 1994.
- [Chr90] J. Christensen. A hierarchical planner that generates its own abstraction hierarchies. In *Proceedings AAAI-90*, pages 1004–1009. AAAI, 1990.
- [CLR91] T. H. Corman, C. E. Leiserson, and R. L. Rivest. *Algorithms*. MIT Press, Cambridge, Massachusetts, 1991.
- [DB88] Thomas Dean and Mark Boddy. Reasoning about partially ordered events. *Artificial Intelligence*, 36(3):375–399, 1988.
- [Dea84] Thomas Dean. Managing time maps. In *Proceedings of the Canadian Society for Computational Studies of Intelligence*. CSCSI, 1984.

- [Eva90] C. Evans. The macro-event calculus: Representing temporal granularity. In *Proceedings of PRICAI'90*, 1990.
- [GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman and Company, New York, 1979.
- [GN91] Naresh Gupta and Dana S. Nau. Complexity results for blocks-world planning. In *Proceedings AAAI-91*, pages 629–633. AAAI, 1991.
- [GS93] A. Gerevini and L. Schubert. Efficient temporal reasoning through timegraphs. In *Proceedings IJCAI-93*. IJCAI, 1993.
- [KG77] Kenneth Kahn and G. Anthony Gorry. Mechanizing temporal knowledge. *Artificial Intelligence*, 9:87–108, 1977.
- [Kno91] Craig A. Knoblock. Search reduction in hierarchical problem solving. In *Proceedings AAAI-91*, pages 686–691. AAAI, 1991.
- [Kor87] Richard Korf. Planning as search: A quantitative approach. *Artificial Intelligence*, 33(1):65–88, 1987.
- [KS86] Robert Kowalski and M. J. Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67–95, 1986.
- [Lan88] Amy L. Lansky. Localized event-based reasoning for multiagent domains. *Computational Intelligence*, 4(4), 1988.
- [LD93] Shieu-Hong Lin and Thomas Dean. Exploiting locality in temporal reasoning. In *Proceedings of the Second European Workshop on Planning*, 1993.
- [MB83] Jitendra Malik and Thomas O. Binford. Reasoning in time and space. In *Proceedings IJCAI 8*, pages 343–345. IJCAI, 1983.
- [Mea92] A. Montanari and et al. Dealing with time granularity in the event calculus. In *Proceedings of FGCS'92*, 1992.
- [NB92] Bernhard Nebel and Christer Bäckström. On the computational complexity of temporal projection and plan validation. In *Proceedings AAAI-92*, pages 748–753, 1992.
- [PS82] Christos H. Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization*. Prentice-Hall, Englewood Cliffs, New Jersey, 1982.
- [Sac74] Earl Sacerdoti. Planning in a hierarchy of abstraction spaces. *Artificial Intelligence*, 7:231–272, 1974.
- [Vil82] Marc Vilain. A system for reasoning about time. In *Proceedings AAAI-82*. AAAI, 1982.
- [YT90] Qiang Yang and Josh D. Tenenber. Abtweak: Abstracting a nonlinear, least commitment planner. In *Proceedings AAAI-90*, pages 204–209. AAAI, 1990.