Localized Temporal Reasoning Using Subgoals and Abstraction

Sheiu-Hong Lin and Thomas Dean

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-94-13
March 1994

Localized Temporal Reasoning Using Subgoals and Abstraction

Shieu-Hong Lin Thomas Dean[1]

Department of Computer Science
Brown University, Providence, RI 02912

[1]This 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.

Abstract

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 immediately following which a goal is achieved. Previous research shows that the associated decision problems are hard even for very restricted cases. In this paper, we present a localized temporal reasoning algorithm which uses subgoals and abstraction to exploit structure in temporal ordering and causal interaction. We investigate (1) locality regarding ordering constraints that group events hierarchically into sets called regions, and (2) locality regarding causal interactions among regions, which is characterized by subsets of the set of conditions. The complexity for a problem instance of temporal reasoning is quantified by the inherent locality in the instance. We theoretically demonstrate the substantial improvements in performance by exploiting locality.

Key words: Temporal Reasoning about Events, Planning, Subgoals, Abstraction

Contents

List of Figures

1 Introduction

We are interested in reasoning about a dynamical system whose state is modeled by a set of conditions $\mathcal{P}$. Each condition has a status, true or false, at a given point in time and the *evolution of the system* corresponds to a sequence of state changes. The evolution of the system depends on the interactions among a set of events $\mathcal{E}$. An event e in $\mathcal{E}$ changes the system state according to a set of *causal rules* associated with e. A causal rule associated with an event e is a STRIPS-like operator describing the causal effects of e when the antecedent requirements of the rule are satisfied. By associating multiple rules with an event, we can model the conditional effects of the events. In this way, each event determines a state-transition relation on the state space determined by $\mathcal{P}$ [LD93]. The ordering of the events is uncertain. The possible event sequences are determined by a given set of ordering constraints on the events. Generally, many event orderings are possible, each of which may result in a different evolution of the system.

Given the initial state and a set of goal conditions $\mathcal{G}$, the task of *temporal projection* is to (1) detect the existence of a possible event sequence immediately following which the conditions in $\mathcal{G}$ have specified statuses, and (2) generate one such event sequence if one exists. In other words, the task of temporal projection is to predict the possibility of reaching the goal states where the conditions in $\mathcal{G}$ have the specified statuses, and report one possible event sequence which ends in the goal states. For example, consider a partial plan where the ordering of the components is constrained by some partial order. Using temporal projection, we can coordinate the component events to reach the goal states in a way consistent with the inherent ordering constraints. On the other hand, we can also apply temporal projection to validate the partial plan by (1) specifying the undesirable states as goal states where the set of goal conditions have the specified statuses, (2) detecting the possibility of reaching undesirable states, and (3) reporting such a possible event sequence as evidence.

Temporal projection can be considered as a special kind of planning. For temporal projection, the planner does not have full control over the events. Instead, the reasoner need to achieve the goal under the restrictions that (1) the number of events is fixed, and (2) the event ordering must be consistent with the inherent ordering constraints. Dean and Boddy [DB88] study a variant of temporal projection. Their investigation focuses on the numbers of preconditions and postconditions in causal rules, and the number of causal rules associated with events. They show that temporal projection is NP-Complete even if we severely restrict the forms of causal rules and the numbers of rules associated with events. Nebel and Bäckström [NB92] prove the NP-Completeness of a special case of temporal projection, which is previously conjectured to be tractable in [DB88]. For this special case, the corresponding planning problem turns out to be solvable in polynomial time [NB92]. This happens because in the corresponding planning problem, the planner is free to select an arbitrary set of event instances from a fixed set of event types with no ordering constraints on events, while in temporal projection, the reasoner does not have full control over events and their ordering. It indicates that ordering constraints play an important role in the computation complexity.

The associated decision problems for temporal projection [DB88] [NB92] [LD93] and planning [BK91] [Byl91] [Cha87] [DB88] [GN91] are hard even for very restricted cases. We have been trying to understand why temporal projection are so difficult and what, if any, structure might be extracted from such problems to expedite decision making and inference [LD93]. We find that the size of the state space is also an important factor of complexity. We depict the complexity trade-offs involving (1) the ordering constraints on events, (2) the size of the state space, and (3) the number of causal rules associated with individual events. Our results indicate that it is necessary to exploit structure to expedite temporal projection. Our investigations have focused on the notions of locality in event ordering and the structure of the search spaces.

In this paper, temporal projection is viewed as search. To expedite search, we develop a localized-reasoning algorithm to exploit structure in problem instances. The use of event-based localized reasoning in planning has been advocated by Lansky in GEM [Lan88]. Localized reasoning exploit locality to improve performance. In GEM, locality regarding domain behavior is modeled by "regions". "Regions" in GEM are composed of sets of interrelated events whose behaviors are delimited by event-based temporal logic constraints. In this paper, we investigate the corresponding notion of regions in the temporal projection problem, which models locality in time. In addition, we study the dependencies between conditions and regions, which models locality in search spaces. Our localized-reasoning algorithm expedites search by subdividing a global search space into local search spaces. Instead of a global search, we conduct local searches and propagate the effects of local searches to derive a solution.

Locality in time in a particular problem instance is modeled by a hierarchy of regions. A region is a subset of closely related events which occur as an atomic group. The events outside a region must occur either before or after the events in the region. Given a hierarchy of regions, we can identify subsets of the conditions which characterize locality in the search space. The coupling conditions of a region R are the media for inter-regional interactions between the events in different child regions of R. The abstract conditions of a region R are the media for the interactions between the events in region R and the events not in R. The knowledge of abstract conditions and coupling conditions enables us to construct local search spaces for individual regions. Locality in a problem instance can be measured by the sizes of the sets of coupling conditions and the numbers of child regions in individual regions. The computation complexity of our algorithm is quantified by the locality measure of the problem instance. Our theoretical results demonstrate the dramatic performance improvement by exploiting locality in temporal projection.

The notions of subgoals and abstract events play important roles in our localized-reasoning algorithm. Our approach of subgoal and abstraction is different from previous work of using subgoal [Kor87] and abstraction [Chr90] [Kno91] [YT90] in reducing the search effort in planning. In this paper, subgoals and abstract events are derived according to the structure in event ordering and search space inherent from the individual problem instances. Given an instance of the temporal projection problem, the goal can be decomposed into subgoals in individual regions. Starting from the bottom level of the region hierarchy, we construct and search the local search spaces of individual regions level by level until we reach the root. In a region R, we conduct local search to achieve the regional subgoal. We also identify the antecedent requirements and the causal effects to achieve the regional subgoal. These requirements and effects are summarized as a set of causal rules regarding the abstract conditions of region R. We refer to this set of causal rules as the abstract event e_R of the region R, and propagate this information to the R's parent region in the region hierarchy. The local search space of region R is constructed according to (1) the information of all the abstract events propagated to R, and (2) the coupling conditions of R. In this way, typically only a small number of abstract events and a small subset of the conditions are involved in the local search of a region. This is the reason why our localized-reasoning algorithm can substantially improve the computation efficiency.

The remainder of the paper is organized as follows. In Section 2, we use an example to illustrate the perspectives and applications of reasoning about the consequences of events. Section 3 starts with the definitions and notations to describe the temporal projection problem and its variants. We then investigate the complexity trade-offs involving the numbers of rules of events, the ordering constraints on events, and the sizes of state spaces. Section 4 illustrates the notions of regions, region hierarchy, and locality in event ordering. Section 5 depicts the locality in causal interaction among regions, and identifies the characteristic subsets of conditions of individual regions. This allows us to define regional subgoals and abstract events of individual regions. In Section 6, we propose our localized-reasoning algorithm to exploit locality in temporal projection. We analyze the complexity of the algorithm and theoretically demonstrate the substantial performance improvement.

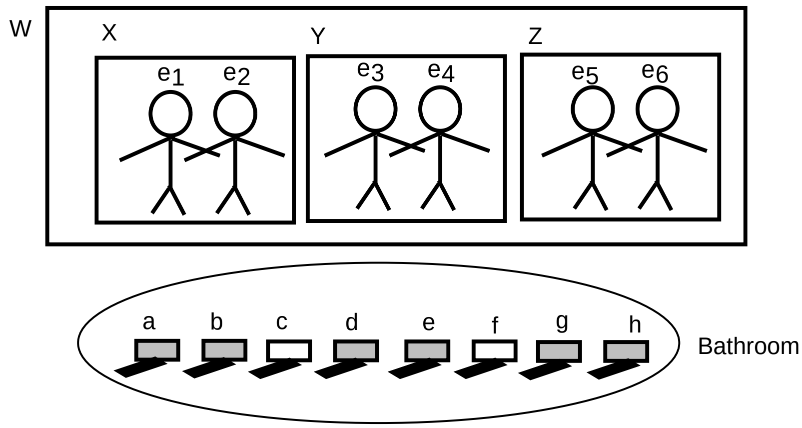
2 Reasoning about the Consequences of Events

2.1 A Bathroom Example

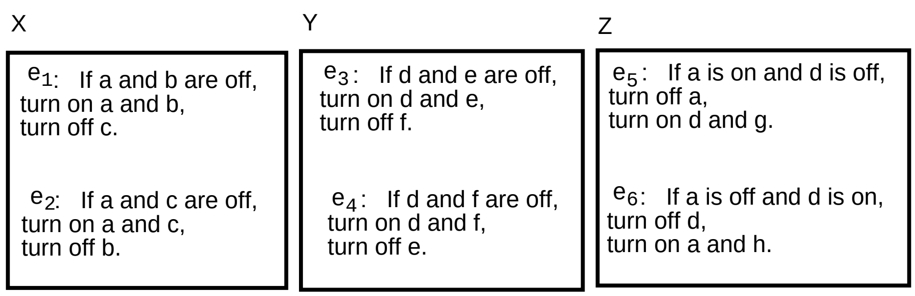
Consider the following example. In Figure 1-a, six tenants, $e_1, e_2, e_3, e_4, e_5, e_6$ share the bathroom of their apartment. e_1 and e_2 are a couple (couple X). e_3 and e_4 are a couple (couple Y). e_5 and e_6 are a couple (couple Z). We refer to these six tenants as a group W. In the bathroom,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

All these six persons work late at night. When they arrive at home, each person use the bathroom once before they go 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 switches a and b and turn off switch c if switches a and b are both off when he enters the bathroom; otherwise, e_1 just let all the switches intact.

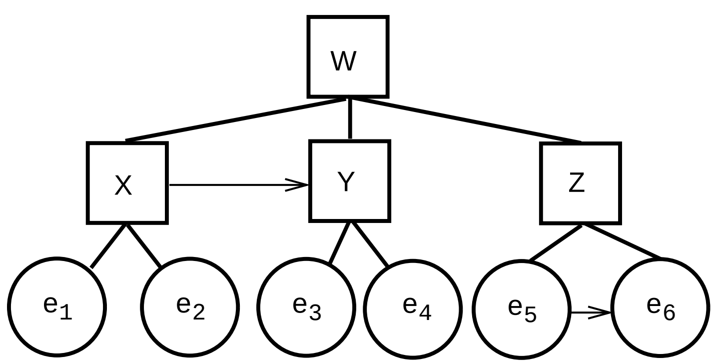
Figure 1-c displays ordering constraints regarding the use of bathroom. For each couple, they use the bathroom in a row. In other words, immediately after one uses the bathroom, the other one will use the bathroom. In addition, couple X always uses the bathroom sometime after couple Y uses



(a)



(b)



(c)

Figure 1: Temporal reasoning about sharing a bathroom

it, while e_6 always uses the bathroom sometime before e_5. The landlord of the apartment uses the bathroom once and goes to sleep before the tenants come back. After using the bathroom, the landlord turns off all the switches. The following four scenarios describe various tasks in reasoning about the possible consequences of using the bathroom.

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 are all turned on overnight. The landlord would like to
(1) determine whether the switches a, b, d, e, g, h may be turned on overnight after all of the tenants use the bathroom, and if this is possible,
(2) come up with a total order on the tenants which ends in the undesirable situation if the tenants use the bathroom in this order, and
(3) explain why the total order and the individual behaviors will result in the undesirable situation.

Scenario 2: To save electricity, the landlord would like to derive a total order on the tenants such that all the switches are automatically turned off after the six tenants use the bathroom in this order.

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 a time. The landlord would like to determine whether such an undesirable situation can occur while one of the tenants is using the bathroom.

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

2.2 Perspectives and Applications

In this bathroom example, we have a description of a dynamical system, and our task is to reason about the consequences of events in this system. This is the essence of the kind of temporal reasoning problem considered in this paper. The following is the description of the underlying dynamical system in the bathroom example.

States: The statuses of the eight switches determine the state of the dynamical system. In the initial state (i.e. the state at the time immediately before the tenants use the bathroom), all the switches are off.

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

Uncertainty in System Evolution: Events occur as a sequence while there is uncertainty in event ordering. Many total orders on the 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 evolution.

We are interested in reasoning about the consequences of the events in such a dynamical system. This is important in a wide variety of applications like planning, plan debugging, scheduling, coordinating multiple agents.

Planning and Plan Debugging: Events together with the ordering constraints on events define a partial plan. Given a partial plan, it is important to detect the potential faults in this partial plan. We would also like to come up with a possible execution in which a fault occurs. This can serve as an evidence and an explanation for the bugs in the plan. For example in Scenarios 1, 3 and 4, we have a partial plan about using the bathroom. A fault is a state where the switches a, b, d, e, h are at the same time. Our task is to predict the possibility of entering a fault state at specific points in time in a possible execution of the plan. In addition, as described in Scenario 1, we also like to derive a trace and explanation of how and why a fault may occur. For this purpose, we can consider fault states as goal states and our task is to derive a total order on events such that we can enter a goal state if the events occur in this order. This task is a special kind of planning.

Scheduling: Given the knowledge about a set of events and their cause-and-effect relationships, it is important to schedule these events so that the system evolution meets our need. For example in Scenario 2, our goal is to save electricity. The goal state is the state in which all the switches are off. Therefore our task is to derive a schedule regarding the uses of the bathroom that ends in the goal state.

Coordinating Multiple Agents: Multiple agents can induce multiple events. The behaviors of the agents can be encoded as causal rules associated with the corresponding events. It is important to coordinate the interactions among multiple agents to achieve a goal. For example in Scenario 2, our task is to coordinate the six tenants to save electricity.

2.3 Definitions and Representations

Before we define the temporal reasoning problem considered in this paper, we first introduce definitions and representation to be used through out this paper.

Definition 1 (Conditions and States) A condition p is a state variable whose status is either true or false at a point in time. A set of conditions $\mathcal{P}$ of size m determines a state space $S_{\mathcal{P}}$. The statuses of the conditions in $\mathcal{P}$ at a point in time determines the state at that time. A state u in $S_{\mathcal{P}}$ is represented as $(x_1, x_2, \ldots, x_m)$ in $\{true, false\}^m$ where x_i indicates the status of condition p_i.

Remark. In the following, we implicitly assume that the state at a point in time is described by a set condition $\mathcal{P}$, and refer to the state space as $S_{\mathcal{P}}$.

Definition 2 (Expression) An expression is a set of condition/status pairs of the form $\langle p, status \rangle$ where p is a condition and $status$ is either true or false. An expression α is true in a state u if and only if for each condition/status pair $\langle p, true \rangle$ ($\langle q, false \rangle$) p is true in u (q 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. A rule $r = \alpha \rightarrow \beta$ is applicable in state u if and only if α is true in state u. Rule r can cause a state transition from state u to state v where (1) rule r is applicable in state u, (2) β is true in state v, and (3) the statuses of the conditions not appearing in β remain the same in v as they are in u.

Remark. A rule $r = \alpha \rightarrow \beta$ represents a STRIPS-like operator. α is the antecedent requirement of rule r and β is the causal effect of rule r. The conditions appearing in α (β) are the preconditions (postconditions) of rule r.

Definition 4 (Events) An event is represented as a set of causal rules. Each event occurs at a distinct point in time. Given the state u immediately before an event e occurs, event e can trigger a state transition from state u to another state v if and only if one of event e's causal rules can cause a state transition from u to v. If two or more rules of event e are applicable in state u, event e triggers state transition nondeterministically according to one of the applicable rules.

Remark. If an event e cause a state transition from state u to state v due to a causal rule r, we say that the rule r is applied by the event e in state u. An event precipitates change of system state. The effects of an event is conditional on (1) the state immediately before the event occurs, and (2) the set of causal rules associated with the event. In this way, an event determines a state-transition relation on $S_{\mathcal{P}} \times S_{\mathcal{P}}$.

Definition 5 (Event Sequences) An event sequence $\mathbf{q}$ of length h over a set of events $\mathcal{E}$ ($h \leq \mathcal{E}$) is a sequence $\langle e_1, e_2, \ldots, e_h \rangle$ where for $1 \leq i \leq h$ $e_i \in \mathcal{E}$ and $e_i \neq e_j$ if $i \neq j$. e_h is the tail of $\mathbf{q}$. A prefix of an event sequence $\mathbf{q} = \langle e_1, e_2, \ldots, e_h \rangle$ is an event sequence $\mathbf{q}' = \langle e_1, e_2, \ldots, e_{h'} \rangle$ where $h' \leq h$.

Remark. When a set of events occur as a sequence, it causes a sequence of state transitions. This corresponds to the evolution of system, which critically depends on (1) the ordering of the events and (2) the choices of applicable rules for individual events.

Definition 6 (Trajectory) A state sequence $\langle s_0, s_1, s_2, \ldots, s_h \rangle$ is a trajectory of an event sequence $\mathbf{q} = \langle e_1, e_2, \ldots, e_h \rangle$ if and only if each event e_i can trigger a state transition from state s_{i-1} to state s_i for $1 \le i \le h$.

Remark. A trajectory of an event sequence records a possible system evolution triggered by the event sequence.

Definition 7 (Sets of Event Sequences) Given a set of ordering constraints $\mathcal{O}$ on a set of events $\mathcal{E}$, $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$is composed of the event sequences over $\mathcal{E}$ which are (1) of length $|\mathcal{E}|$ and (2) consistent with the ordering constraints in $\mathcal{O}$.

$\overline{\mathbf{Q}}^{\mathcal{E}}_{\mathcal{O}}$=$\{\mathbf{q}' \mid \mathbf{q}'$ is a prefix of an event sequence $\mathbf{q} \in \mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}\}$.

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

Remark. $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$is composed of all the total orders on $\mathcal{E}$ which are consistent with the ordering constraints $\mathcal{O}$. $\overline{\mathbf{Q}}^{\mathcal{E}}_{\mathcal{O}}$is composed of the prefix of the event sequences in $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$. $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O},e}$is composed of the event sequences in $\overline{\mathbf{Q}}^{\mathcal{E}}_{\mathcal{O}}$whose tail event is e.

3 Temporal Projection

3.1 The Temporal Projection Problem

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

- $\mathcal{P}$ is a set of conditions that model the system states;
- s_0 is the initial state;
- $\mathcal{G}$ is the goal, which is an expression;
- $\mathcal{E}$ is a set of events;
- $\mathcal{O}$ is a set of ordering constraints on $\mathcal{E}$.

Definition 8 (Goal States) A state u in $S_{\mathcal{P}}$ is a goal state if and only if the goal $\mathcal{G}$ is true in state u.

e_1={ (<a,true>,<b,true>) $\Rightarrow$ (<a,false>,<b,false>,<c,true>) ,
(<a,false>) $\Rightarrow$ ( )), (<b,false>) $\Rightarrow$ ( ))
}

e_2={ (<a,true>,<c,true>) $\Rightarrow$ (<a,false>,<c,false>,<b,true> ,
(<a,false>) $\Rightarrow$ ( )), (<c,false>) $\Rightarrow$ ( ))
}

e_3 ={ (<d,true>,<e,true>) (<d,false>,<e,false>,<f,true>) ,
(<d,false>) $\Rightarrow$ ( )), (<e,false>) $\Rightarrow$ ( ))
}

e_4 ={ (<d,true>,<f,true>) $\Rightarrow$ (<d,false>,<f,false>,<e,true>) ,
(<d,false>) $\Rightarrow$ ( )), (<f,false>) $\Rightarrow$ ( ))
}

e_5={ (<a,false>,<d,true>) $\Rightarrow$ (<a,true>,<d,false>,<g,false>) ,
(<a,true>) $\Rightarrow$ ( )), (<d,false>) $\Rightarrow$ ( ))
}

e_6={ (<a,true>,<d,false>) $\Rightarrow$ (<a,false>,<d,true>,<h,false>) ,
(<a,false>) $\Rightarrow$ ( )), (<d,true>) $\Rightarrow$ ( ))
}

Figure 2: The events and causal rules in the bathroom example

Definition 9 (Temporal Projection) Given an instance of the temporal projection problem, the PRJ task of temporal projection is to (1) determine the existence of event sequences in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$immediately following which we may enter goal states, (2) generate such an event sequence $\mathbf{q}$ if one exists, and (3) determine the rules applied in the individual events of $\mathbf{q}$ to enter a goal state.

Example 1 The following is a problem instance and its associated task of temporal projection, which corresponds to Scenario 1 in the bathroom example.

- $\mathcal{P} = \{a, b, c, d, e, f, g, h\}$, which corresponds to the eight switches in the bathroom. A condition is *true* if and only if the corresponding switch is *off*.

- In the initial state s_0, all the conditions are initially true, since the landlord turns off all switches before the tenants use the bathroom.

- $\mathcal{G} = (\langle a, false\rangle, \langle b, false\rangle, \langle d, false\rangle, \langle e, false\rangle, \langle g, false\rangle, \langle h, false\rangle)$.

- $\mathcal{E} = \{e_1, e_2, e_3, e_4, e_5, e_6\}$, which corresponds to the uses of bathroom by the six tenants. Figure 2 display the causal rules associated with the individual events.

- Figure 1-c depicts the ordering constraints $\mathcal{O}$ on $\mathcal{E}$. (For notational convenience, we use W as an alias name for $\mathcal{E}$.) 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.

- Our task is to (1) determine the existence of a schedule $\mathbf{q}$ in $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$regarding the uses of bathroom by the tenants such that the switches a, b, d, e, g, h are on (i.e. $\mathcal{G}$ is true) immediately after schedule $\mathbf{q}$ is executed, (2) generate such a schedule $\mathbf{q}$ if one exists, and (3) determine the rules applied by the individual tenants in schedule $\mathbf{q}$ which make $\mathcal{G}$ true.

Definition 10 (Variants of Temporal Projection) Given an instance of the temporal projection problem, the PRJ1 task [2] and the PRJ2 task [3] are to determine the existence of event sequences in $\overline{\mathbf{Q}}^{\mathcal{E}}_{\mathcal{O}}$and $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O},e}$respectively (e is a specified event) immediately following which we may enter goal states.

Theorem 1 The three variants of the temporal temporal projection problem, namely PRJ, PRJ1, and PRJ2, can be reduced to one another in polynomial time. [4]

3.1.1 Standard Form of Temporal Projection

Temporal projection as defined in Definition 9 obviously covers the kind of decision task described in Scenarios 1 and 2 in Section 2.1, which is referred to as the PRJ task. The two variants of temporal projection, the PRJ1 task and PRJ2 task, correspond to the reasoning tasks described in Scenarios 3 and 4 in Section 2.1 respectively. Theorem 1 shows that these three variants, PRJ, PRJ1, PRJ2 are computationally equivalent. In the remainder of this paper, we adopt Definition 9 for the PRJ task as the standard form of all the reasoning tasks considered through out this paper.

[2]PRJ1 is a variant of the temporal projection problem which is considered by Lin and Dean in [LD93].

[3]PRJ2 is a variant of the temporal projection problem which is considered by Dean and Boddy in [DB88] and Nebel and Bäckström in [NB92].

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

3.2 Temporal Projection as Search

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

Time and State: The set of vertices V represents possible system states at different points in time. Therefore each vertex contains two components: (1) a system state, which is specified by the statuses of the conditions in $\mathcal{P}$, and (2) a time point, which is determined by whether the individual events in $\mathcal{E}$ have occurred or not.

Occurrences of Events: The set of arcs A represents the possible occurrences of events at different points in time. We construct an arc (u, v) in A if (1) by applying a causal rule r, event e can cause a state transition from the state in vertex u to the state in vertex v, (2) e can occur immediately after those events that are marked as occurred in vertex u without violating the ordering constraints in $\mathcal{O}$, and (3) in vertex u, e has not yet occurred while in vertex v e has occurred. A *rule-tag* (e, r) on such an arc (u, v) to indicate that by applying rule r, event e can cause a state transition from the state in vertex u to the state 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 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 individual events to enter the goal state.

The following procedure **TemporalSearch** provides more details of temporal projection as search [5].

Procedure TemporalSearch

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

output: (1) a set F of the goal states reachable immediately after the 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 individual events in $\mathbf{q}$ to enter the goal state f;

(4) a trajectory $\mathbf{t_q}$ of $\mathbf{q}$ that ends in the goal state f.

1. Partition the set of events $\mathcal{E}$ into m chains where the ith ($1 \leq i \leq m$) 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 maps state s to a specific point in time. A vertex t in V is of the form $(s, x_1, x_2, \ldots, x_m)$. 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 time counter for the ith chain; x_i indicates that the first x_i events in the ith chain have occurred while the remaining events in the ith 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 $(s, x_1, x_2, \ldots, x_k - 1, \ldots, x_m)$ to
   $(t, x_1, x_2, \ldots, x_k, \ldots, x_m)$ if (1) e is the x_kth event in the kth 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 kth chain and the first x_i events in the ith chain ($1 \leq i \leq m, i \neq k$) without violating the ordering constraints in $\mathcal{O}$.
   We add a rule-tag (e, r) to indicates that event e can trigger such a state transition by applying causal rule r. In other words, each arc models the possible occurrence. These arcs comprise the arc set V of G_I.

4. In the following, we reduce the task of temporal projection to graph reachability. We can apply standard graph-reachability algorithms to derive the solutions.

   - Search the graph G_I. A goal state t can be reached immediate after an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$if and only if the goal vertex $(t, l_1, l_2, \ldots, l_m)$

[5] In Section 6, this procedure is adapted for local reasoning in local search spaces.

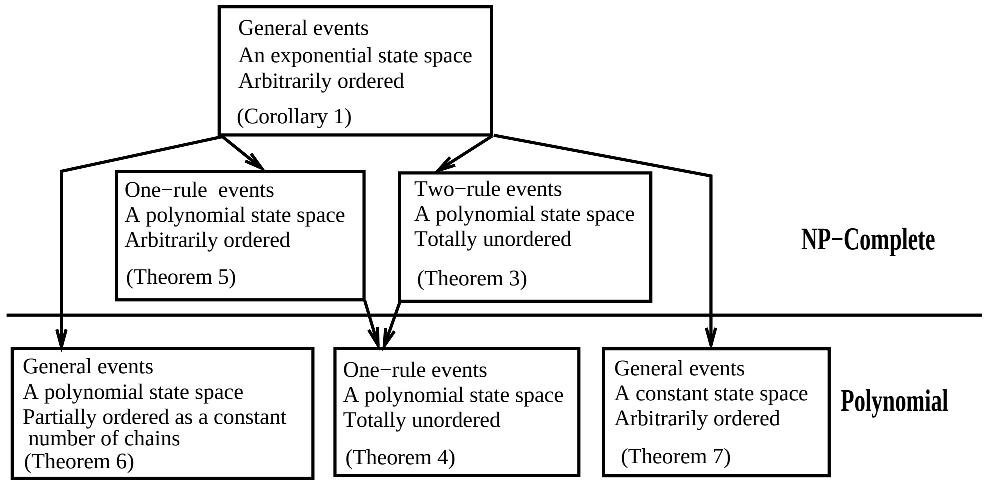


Figure 3: Complexity trade-offs of temporal projection

is reachable from the root vertex $(s, 0, 0, \ldots, 0)$ in graph G_I. Add such a state t to the set F.

- Search for a directed path from the root vertex $(s, 0, 0, \ldots, 0)$ to a goal vertex
$(f, l_1, l_2, \ldots, l_m)$. 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 2 The search graph $G_I = (V, E)$ constructed in procedure TemporalSearch 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 TemporalSearch runs in $O(2^{2(|\mathcal{P}|+|\mathcal{E}|)})$ time.

3.3 Factors of Computational Complexity

In the remainder of this section, we show that (1) the number of causal rules associated with individual events, (2) the ordering constraints on events, and (3) the size of the state space all contribute to the computation complexity. First we define the following terms.

Definition 11 (One-rule Events and Two-rule Events) A one-rule event is associated with a single rule which is applicable in exactly one state and map the state to some other state. A two-rule event is associated with two rules, each of which is applicable in exactly one state and map the state to some other state.

Remark. A one-rule event is applicable in only one state. A two-rule event is applicable in two states. A general event can have multiple rules, each of which can be applicable in many states.

Definition 12 (Partially Ordered as Chains) We say that events are partially ordered as m chains if the events can be partitioned into m disjoint event subsets such that the events in each subset are totally ordered.

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

- events are categorized as *one-rule*, *two-rule* or *general* according to how they trigger state transitions;
- ordering constraints on events are classified as one of (1) partially ordered as a constant number of chains, (2) totally unordered (no ordering constraints), or (3) arbitrarily ordered;
- the sizes of the state spaces are categorized as one of constant, polynomial, or exponential, in the size of the event set $\mathcal{E}$ (In other words, $|\mathcal{P}| = O(1)$, $|\mathcal{P}| = O(\log |\mathcal{E}|)$, $|\mathcal{P}| = O(|\mathcal{E}|)$ respectively);

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 whether a goal state is reached immediately following $\mathbf{q}$ in polynomial time. Therefore we have the following lemma.

Lemma 1 The temporal projection problem is in NP.

The following theorems indicate that we have very limited ability to deal with primitive events in a state space of polynomial size if the events can occur in arbitrary order.

Theorem 3 Temporal projection regarding two-rule events, totally unordered, in a state space of polynomial size is NP-Complete.

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

Theorem 5 Temporal projection regarding one-rule events, arbitrarily ordered, in a polynomial-size state space is NP-Complete. [6]

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

The following two theorems indicate that small state space and structure in event ordering can expedite temporal projection.

Theorem 6 Temporal projection regarding events partially ordered as $O(1)$ number of chains of length $O(|\mathcal{E}|)$ and $O(\log|\mathcal{E}|)$ number of chains of $O(1)$ length in a polynomial-size state space can be solved in polynomial time.

Theorem 7 Temporal projection regarding general events, arbitrarily ordered, in a constant-size state space is solvable in polynomial time.

4 Locality in Event Ordering

Theorem 3 and Theorem 5 show that temporal projection is NP-Complete if events can occur in arbitrary order even in a polynomial-size state space. On the other hand, Theorem 6 and Theorem 7 demonstrate that structure in event ordering and a small state space can reduce the complexity of temporal projection. These complexity results motivate our effort to exploit inherent locality in event ordering and the dependencies among conditions and events.

When totally unordered, events can occur in arbitrary order. However, there is locality in event ordering if the occurrences of events or subsets of events closely relate to one another. A subset of events may always occur as an atomic unit; all the other events either occur before or after the subset of

[6] Nebel and Bäckström [NB92] proves the NP-Completeness of a special case of the temporal projection problem, which is adapted here to prove Theorem 5.

events. Similarly, several atomic units may always occur as an atomic unit. In this way, temporal locality 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 such kind of temporal locality. In this paper, we investigate locality in event ordering that can be modeled by a *region hierarchy* and *regional ordering constraints*.

Definition 13 (Regions) Given a set of events $\mathcal{E}$ and a set of ordering constraints $\mathcal{O}$ on $\mathcal{E}$, an event subset X of $\mathcal{E}$ is a region in $\mathcal{E}$ if and only if the events in X occur as an atomic group while the events not in X occurring either before or after the events in X. The sets $\mathcal{E}$ and $\{e\}$ for an event e in $\mathcal{E}$ are regions by themselves.

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

Definition 14 (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 (ancestor) region of a region Y if and only if X is a child (parent) region of Y or Y's descendant (parent) region.

Definition 15 (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, $X = \{e_1, e_2\}$, $Y = \{e_3, e_4\}$,
$Z = \{e_5, e_6\}$, are the child regions of $W = X \bigcup Y \bigcup Z$. $\{e_1\}, \{e_2\}$ are the child regions of X. $\{e_3\}, \{e_4\}$ are the child regions of Y. $\{e_5\}, \{e_6\}$ are the child regions of Z. Together, they form a region hierarchy, which is depicted in Figure 1-c as a tree.

Definition 16 (Regional Ordering Constraints) The set of regional ordering constraint $\mathcal{O}_R$ on a region R is a two-tuple $\langle \sigma, \prec \rangle$, where $\sigma = \{R_1, R_2, \ldots\}$ 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 $\mathcal{O}_W = \langle\{X,Y,Z\},\{Y \prec X\}\rangle$.

4.1 Describing Temporal Locality in Task Networks

A region hierarchy and regional ordering constraints can describe the possible executions of the following kind of task networks:

- Tasks are organized as a hierarchy, where an individual task can be recursively decomposed into subtasks. The subtasks of an individual task may be constrained by a partial order. A primitive task in the task network has conditional effects on the system state, which is modeled by a set of STRIPS-like operators.

- For a given task, we can execute its subtasks in any order consistent with the inherent partial order on these subtasks. Once we begin a subtask of the task, we commit to completing all of the primitive tasks associated with that subtask before switching to any other subtasks.

The notions of primitive tasks, tasks, subtasks, task networks, and partial orders on subtasks here correspond to the notions of events, regions, child regions, region hierarchies, and regional ordering constraints respectively. For example, Figure 1-c displays a three-level task network. The root task W is decomposed into three subtasks X, Y, and Z, each of which is further decomposed into two primitive tasks. Task X is constrained to precede task Y, and task e_6 is constrained to precede e_5.

5 Locality in Causal Interactions

5.1 Causal Dependency and Problem Decomposition

5.1.1 Causal Dependency

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 in region R. This can result in significant computational savings. In the following, our investigation focuses on the causal dependencies among regions and subsets of conditions.

Definition 17 (Causal Dependency) 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 e if p is dependent on an event in R.

5.1.2 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 decomposed the problem instance into two independent parts: one with region X and conditions a, b, and c; the other one with region Y and conditions d, e, and f. However, through conditions a and d respectively, region Z interacts with regions X and Y respectively. This complicates the causal interactions among these three regions, since the reasoning tasks in regions X, Y, and Z are not fully independent.

In general, the reasoning tasks in disjoint regions are not fully independent. In these cases, problem decomposition must be achieved by carefully exploiting structure in the causal interactions among regions. In the remainder of this section, we investigate the *regional subgoals* and the *abstractions* of individual regions. Using subgoals and abstractions, temporal projection as search can be decomposed into a sequence of local searches in individual regions. This yields the localized-reasoning algorithm in Section 6.

5.2 Regional Subgoals and Problem Decomposition

Given a region hierarchy, we can identify the local conditions and the subgoal conditions of each individual region. Based on this knowledge, we depicts the regional subgoals of individual regions.

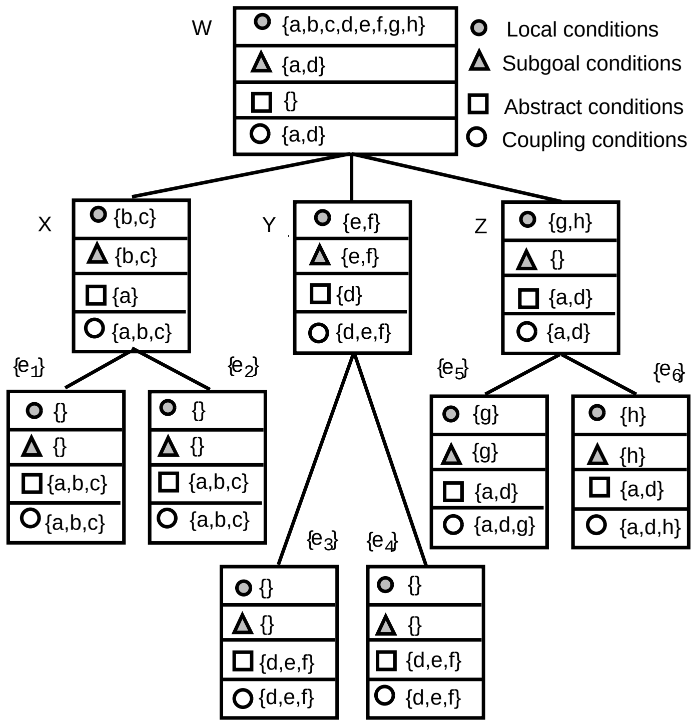


Figure 4: The condition subsets to characterize the causal interactions among regions

Definition 18 (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 19 (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 individual regions in the bathroom example.

The following two 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 status of p can only be locally affected by the events in R. The status of p immediately before the events in R occur is the same as p's initial status. Immediately after all events in R have occurred, the status of p will not be changed any more.

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

In the following, we define the regional subgoals and the incremental subgoals of individual regions.

Definition 20 (Regional Subgoals) Given a goal $\mathcal{G}$, the regional subgoal $\mathcal{G}_R$ of a region R is composed of the condition/status pairs in $\mathcal{G}$ whose condition components are the local conditions of region R. The incremental subgoal $\tilde{\mathcal{G}}_R$ of region R is composed of the condition/status pairs in $\mathcal{G}$ whose condition components are the subgoal conditions of region R.

Since the local conditions of a region R is the union of the local conditions of R's child regions and the subgoal conditions of R, we have the following lemma.

Lemma 4 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.

Theorem 8 The goal $\mathcal{G}$ is true after all events occur if and only if for each region R, the regional subgoal $\mathcal{G}_R$ of R is true immediately after the events in R have all occurred.

By Theorem 8 and Lemma 4, we have the following corollary.

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

Example 6 Figure 5 depicts the incremental subgoals and the regional subgoals in the problem instance about the bathroom example. For example, the regional subgoal of Y is $(\langle e, false \rangle)$, which indicates that the status of condition e must be false immediately after the events in Y have all occurred. The goal $\mathcal{G}$ is equal to the regional subgoal $\mathcal{G}_W$ of the root region W, which contains all the 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 $\tilde{\mathcal{G}}_W$ must be achieved immediately after all events have occurred.

5.2.1 Localized Reasoning Using Regional Subgoals

By Theorem 8 and Corollary 2, temporal projection can be decomposed into a sequence of local reasoning tasks, in each of which we are concerned with achieving the incremental subgoal of an individual region. The following is a localized-reasoning framework for temporal projection. The details are provided in Section 6.

1. *Propagating Localized Reasoning*: Starting from the bottom level of the region hierarchy, we conduct localized reasoning described in step 2 in the individual regions at the same level, and propagate the derived information to their parent regions. We proceed level by level until we finish the localized reasoning in the root region of the region hierarchy.

2. *Deriving Requirements to Achieve Regional Subgoal*: According to the information propagated to region R, we reason about the causal requirements to achieve the regional subgoal $\mathcal{G}_R$ of R. If the incremental subgoal $\tilde{\mathcal{G}}_R$ can not be achieved, we stop and report failure; otherwise, the derived requirements is propagated to the parent region of R.

3. *Generate an Event Sequence to Achieve the Goal*: Using the derived causal requirements to achieve regional subgoals, we can recursively generate the ordering of the child regions of individual regions, which describes an event sequence to achieve the goal $\mathcal{G}$.

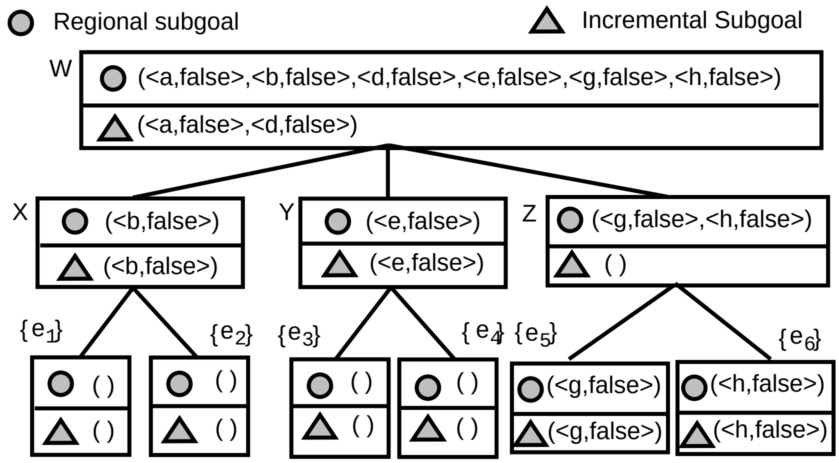


Figure 5: The regional subgoals and incremental subgoals of individual regions

5.3 The Causal Abstraction of a Region

Given a region hierarchy, we can identify the abstract conditions of each individual region. Based on this knowledge, we describe the abstractions of individual regions, which encode the causal requirements to achieve the regional subgoals of individual regions. Figure 4 displays the abstract conditions of individual regions in the bathroom example.

Definition 21 (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 statuses 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, e_2 in region X interact with the other events 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 interactions between the events in R and the events not in R. If region R does not have any abstract conditions, all the conditions involved in the events of R are local conditions of R. In this case, the events in R together with their local conditions can be separated from the other events and conditions to form an independent problem instance.

5.3.1 Abstract Conditions and Regional Subgoals

The effects of the events in a region R can only affect and be affected by the the local conditions and the abstract conditions of R. Since the statuses of the local conditions of R immediately before the events in R occur is the same as their initial statuses, the system evolution triggered by the events in a region R are determined by two factors: (1) the statuses of the abstract conditions of R immediately before the events in R occur, and (2) the ordering of the events in R. Therefore, to achieve the regional subgoals $\mathcal{G}_R$, (1) the set of abstract conditions of R must have appropriate statuses 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 statuses of the abstract conditions of R after achieving the regional subgoals $\mathcal{G}_R$. These cause-and-effect relationships provides an abstraction of region R.

Definition 22 (Abstract States) Given a subset $\mathcal{A}$ of the set of conditions $\mathcal{P}$, $\mathcal{A}$ determines an abstract state space $S_\mathcal{A}$. The statuses 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 an expression α where $\alpha = \{\langle p, status\rangle \mid p \in \mathcal{A}$, $status$ is the status of $p\}$.

Definition 23 (Abstraction of a region) Consider a region R where $\mathcal{A}$ is the set of abstract conditions of R, and $\mathcal{O}$ is the set ordering constraints. An abstraction of a region R is a relation Δ_R 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 $\mathbf{Q}^R_\mathcal{O}$ such that given u as the abstract state of $\mathcal{A}$ immediately before the events in R occur, immediately following $\mathbf{q}$ the regional subgoal $\mathcal{G}_R$ can be achieved and the abstract state of $\mathcal{A}$ is v.

Definition 24 (Abstract Events) An abstract event e_R of a region R is a set of causal rules where (1) the preconditions and postconditions of the rules in e_R are abstract conditions of R, and (2) e_R determines the abstraction relation Δ_R of R.

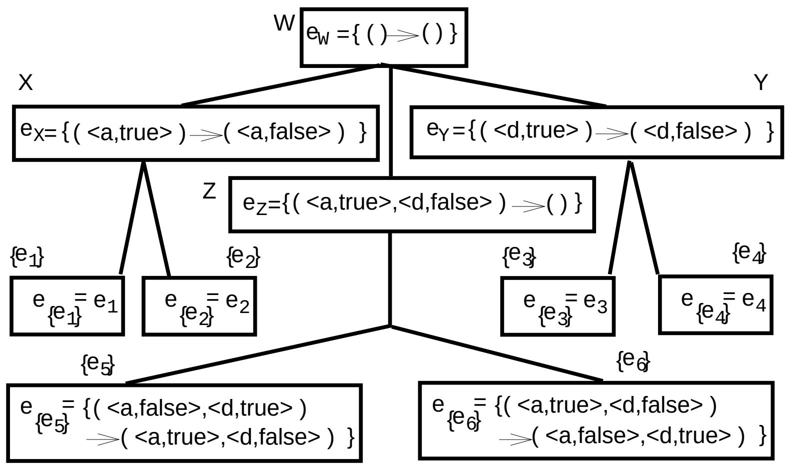


Figure 6: The abstract events of individual regions

Remark. The abstraction relation Δ_R provides the information regarding the antecedent requirements and consequent effects on the abstract conditions of R to achieve the regional subgoals $\mathcal{G}_R$. An abstract event e_R is a representation of Δ_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 6.

Example 7 Figure 6 displays the abstract events of the regions in the region hierarchy in the bathroom example. At the top level, the global region W has no abstract conditions. Therefore $(\langle\rangle \rightarrow \langle\rangle)$ is the only rule in the abstract event of the root region. At the second level, it is not hard to see that a (d) must be true immediately 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. Similarly, we can derive the abstract event e_Z for region Z as described in Figure 6.

At the bottom level, except for regions $\{e_5\}$ and $\{e_6\}$ each region has an empty regional subgoal, in which the only requirements to achieve the regional subgoal is inherent from the causal rules associated with 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 $(\langle a, true \rangle, \langle d, false \rangle) \rightarrow (\langle a, false \rangle, \langle d, true \rangle)$ to make h false. Consequently, a must become false and d must become true immediately after e_6 occurs. Therefore, the abstract state of $\mathcal{A} = \{a, d\}$ immediately before the event sequence $\langle e_6 \rangle$ occur must be $\{\langle a, true \rangle, \langle d, false \rangle\}$, while the abstract state of $\mathcal{A} = \{a, d\}$ immediately after the event sequence $\langle e_6 \rangle$ occur must be $\{\langle a, false \rangle, \langle d, true \rangle\}$. 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 $e_{\{e_5\}}$ for region $\{e_5\}$ as described in Figure 6.

6 Localized Temporal Reasoning Using Subgoals and Abstractions

In this section, we exploit the structure in a region hierarchy to expedite temporal projection. Using the notions of regional subgoals and abstractions of regions, temporal projection as search is reduced to a sequence of local searches in individual regions.

6.1 Coupling Conditions and Local Reasoning

Definition 25 (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 bath room example. For example, the three child regions X, Y, and Z of region W interact with one another through the coupling conditions a, d of region W.

Theorem 9 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 set of coupling conditions of a region R plays an important role about the local reasoning in R. Instead of reasoning about the entire state space, we only need to focus on the abstract state space determined by the coupling conditions of R, which results in significant search reduction.

- The child regions of a region R is coupled together through the coupling conditions of R. The interactions among these child regions are precisely described by the interactions among the abstract events of these child regions, whose union is the set of coupling conditions of R. Therefore the coupling conditions are the media for the inter-regional interactions among the child regions of R.

- In order to derive an abstraction of region R, we need to reason about the statuses of the set of abstract conditions and the set of subgoal conditions, whose union is the set of coupling conditions according to Theorem 9. specified in the regional subgoal.

6.2 A Localized Temporal Reasoning Algorithm

In the following, we present a localized temporal reasoning algorithm for temporal projection. Procedure Localized-Reasoning control the propagation of local reasoning in individual regions. Procedure Region-Abstraction and procedure Generate-Sequence handle the local reasoning tasks in a region, both of which employ the procedure TemporalSearch described in Section 3.3 to conduct local search. Procedure Localized-Reasoning guarantees that the following causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ is available when we perform the local reasoning tasks in a region R 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;

- $\tilde{\mathcal{G}}_R$ 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 R.

Algorithm Localized-Reasoning
input: a problem instance $\langle \mathcal{P}, s_0, \mathcal{G}, \mathcal{E}, \mathcal{O} \rangle$.
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 sequences exists;
otherwise, report failure in finding such a sequence.

1. Derive the following causal knowledge by examining 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 depicted Figure 5.

2. Starting from the bottom level of the region hierarchy, we call procedure Region-Abstraction for the regions at the same level respectively, and then proceed in the same way level by level until we reach the root region of the region hierarchy.

3. For each region R, the procedure Region-Abstraction derives the abstract event e_R as the causal requirements to achieve the regional subgoal of R. If the regional subgoal can not be achieved, abort and report failure to achieve the goal $\mathcal{G}$; otherwise, propagate the knowledge of the abstract event e_R to R's parent region.

4. After finish the local reasoning in the root region of the region hierarchy, call procedure Generate-Sequence to generate a solution event sequence $\mathbf{q}$ and the rules applied by the events in $\mathbf{q}$ such that immediately following $\mathbf{q}$ we enter a goal state.

Procedure Region-Abstraction derives an abstract event of a region, which determines the causal requirements to achieve the incremental subgoal of the region.

Procedure Region-Abstraction
Input: the causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$.
Output: if the incremental subgoal of R can be achieved, report an abstract event e_R;
otherwise, stop and report failure in achieving the regional subgoal.

1. Determine the set S of the possible abstract states of $\mathcal{C}_R$ immediately before the events in R occur, whose statuses of the subgoal conditions in $\mathcal{U}_R$ are the same as those in the initial state s_0.

2. For each abstract state s in S, we construct a new instance of the temporal projection problem $\langle \mathcal{C}_R, s, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ regarding the set of coupling conditions $\mathcal{C}_R$, the possible abstract state s, the incremental subgoal $\tilde{\mathcal{G}}_R$, the set of abstract events $\mathcal{X}_R$, and the set of regional ordering constraints $\mathcal{O}_R$.

3. For each new instance $\langle \mathcal{C}_R, s, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$, we call procedure Temporal-Search to derive the set of reachable goal states F of this new instance. Add the pairs (s, f) for each f in F into a set Δ_R.

4. If the set Δ_R is empty, we report failure to achieve the incremental subgoal. Otherwise, for each pair of abstract states (s, f) in Δ_R, we associate a causal rule $\alpha \rightarrow \beta$ with the abstract event e_R where α (β) is the abstract state of $\mathcal{A}_R$ in s (f).

Procedure Generate-Sequence generates an event sequence to achieve the regional subgoal of a region.

Procedure Generate-Sequence
Input: (1) the causal knowledge $\langle \mathcal{A}_R, \mathcal{U}_R, \mathcal{C}_R, s_0, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$;
(2) a pair of abstract states (s', f') of $\mathcal{A}_R$ where s' (f') is the abstract state of $\mathcal{A}_R$ immediately before (after) the events in R occur.
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 regional subgoals $\mathcal{G}_R$ and the abstract state f' is true.

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

2. Construct a new instance of the temporal projection problem $\langle \mathcal{C}_R, s, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$ regarding the set of coupling conditions $\mathcal{C}_R$, the possible abstract state s immediately before the events in R occur, the incremental subgoal $\tilde{\mathcal{G}}_R$, the set of abstract events $\mathcal{X}_R$, and the set of regional ordering constraints $\mathcal{O}_R$.

3. For the new instance $\langle \mathcal{C}_R, s, \tilde{\mathcal{G}}_R, \mathcal{X}_R, \mathcal{O}_R \rangle$, call procedure Temporal-Search to derive an event sequence $\mathbf{q}'$ in $\mathbf{Q}_{\mathcal{O}_R}^{X_R}$ that ends in a reachable goal state f where (i) the incremental subgoal $\tilde{\mathcal{G}}_R$ is true in f, and (ii) the abstract state f' of $\mathcal{A}_R$ is true in f.
Call procedure Temporal-Search to derive (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 $\mathbf{q}'$ to result in the trajectory $\mathbf{t}_{\mathbf{q}'}$.

4. Consider the abstract event of a child region R' as the representative of R'. Determine the total order on the child regions of R according to the sequence $\mathbf{q}'$.
For each child region R' of R, according to the trajectory $\mathbf{t}_{\mathbf{q}'}$ of $\mathbf{q}'$ derived in Step 3 determine the pair of abstract states $(\tilde{s}, \tilde{f})$ of the set of abstract conditions of R' where $\tilde{s}$ ($\tilde{f}$) is true immediately before (after) the abstract event of R' occurs in $\mathbf{q}'$.

5. For each child region R' of R, (i) prepare the pair of abstract states $(\tilde{s}, \tilde{f})$ derived in Step 4 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. Determine the rule applied by an event e in $\mathbf{q}$ according to the rule applied by e in Step 3 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 reasoning algorithm in solving our example problem instance in Figure 1. First, we derive the following knowledge by examining the problem instance: (1) the region hierarchy as depicted in Figure 4-c; (2) the local conditions, the subgoal conditions, the abstract conditions and the coupling conditions as depicted in Figure 4; and (3) the regional subgoals and incremental subgoals of individual regions depicted Figure 5.

Second, starting from the bottom level of the region hierarchy, we call procedure Region-Abstraction to derive the abstractions of individual 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 abstraction for every region, there exists an event sequence $\mathbf{q}$ in $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$to achieve the goal.

Finally, we call procedure Generate-Sequence to recursively derive an event sequence $\mathbf{q}$ in $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$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 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 $\mathbf{Q}^{\mathcal{E}}_{\mathcal{O}}$to achieve the goal. The causal rules applied by the events in $\mathbf{q}$ as a sequence are the first, the second, the first, the first, the first, and the second rule of the events respectively.

6.3 Quantifying the Computation Efficiency

As indicated in Figure 3, temporal projection is NP-Complete even if the number of rules in an event and the size of the state space are severely restricted. Our localized-reasoning algorithm exploits useful structure embedded in individual problem instances to expedite temporal projection. In the following, we define the interaction measure of a problem instance, which determines the computational efficiency of procedure Localized-Reasoning. This allows us to theoretically demonstrate the dramatic improvements in performance by exploiting locality.

Definition 26 (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 $\max_R(b_R + c_R)$.

Remark. The interaction measure $\mathcal{L}$ indicates the magnitude of the causal 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 causal 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.

Remark. 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 under our expectation, 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 case as shown in Theorem 3.

Remark. Our localized reasoning algorithm exploit locality inherent in problem instances to bring in computation 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 dramatically expedite temporal projection. The following two corollaries of Theorem 11 demonstrate the computation savings 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 the situation that (1) every region has $O(1)$ ($O(\log n)$) number of child regions and $O(1)$ ($O(\log n)$) number of abstract conditions and (2) for each region the child regions of the region are totally unordered. In this case, there are $\Omega(n^{(\log n)/2})$ ($\Omega(\log n^{(\log n \times \log\log n)/2})$) possible total orders on events, where n is size of the event set $\mathcal{E}$. It is hard to examine this super-polynomial number of total orders one by one for a solution.

However, the interaction measure *calL* in this case is of $O(1)$ ($O(\log n)$) magnitude. This is because the coupling factor of a region is no more than the sum over the numbers of the abstract conditions of R's child regions according to Theorem 9. Therefore procedure Localized-Reasoning can derive a solution in linear (polynomial) time, which is a dramatic computational saving.

7 Conclusion

We investigate the locality in (1) the ordering constraints that group events hierarchically into sets called *regions*, and (2) the dependencies among conditions and regions. This enables us to describe subgoals and abstractions for individual regions. We develop a localized temporal reasoning algorithm to exploit locality, in which temporal projection as search is reduced to a sequence of local searches in local search spaces. We theoretically demonstrate the dramatic improvements in performance by exploiting locality.

A Proofs

Theorem 1 The decision versions of the three variants of the temporal projection problem, namely PRJ, PRJ1, and PRJ2, can be reduced to one another in polynomial time.

Proof. Given an instance I for the PRJ variant, we can construct the following instance I' for the PRJ2 variant. 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 occur if and only if the new goal in instance I' can be true immediately after event e occur.

Given an instance I for the PRJ2 variant, we can construct the following instance I' for the PRJ1 variant. Suppose e is the specified event in the PRJ2 variant. 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 occurs if and only if the new goal in instance I' can be true immediately after an arbitrary event occurs.

Given an instance I for the PRJ1 variant, we can construct the following instance I' for the PRJ variant. We add a new condition p. All the 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 after an arbitrary event occurs if and only if the new goal in instance I' can be true after all events occur. ■

Theorem 2 The search graph $G_I = (V, E)$ constructed in procedure TemporalSearch 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 TemporalSearch 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. There is no directed path from u to u where the time counters are the same in the head and the tail, since at least one of the time 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 $|S_{calP} \times \prod_{1 \leq i \leq m}(1 + l_i)= 2^{|\mathcal{P}|+|\mathcal{E}|}$ vertices, and at most $2^{2(|\mathcal{P}|+|\mathcal{E}|)}$ arcs. Therefore the procedure TemporalSearch can run in $O(2^{2(|\mathcal{P}|+|\mathcal{E}|)})$ time. ■

Theorem 3 Temporal projection regarding two-rule events, totally unordered, in a state space of polynomial size 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 the 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 PRJ1. (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 cause a state transition from u to v. (3) Each forbidden pair of arcs are mapped to a distinct two-rule event which is associated with the two causal rules corresponding to the two arcs. 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 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 prove the NP-Completeness of this special case. ■

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

Proof. For this special case, the task in the PRJ1 variant of temporal projection can be accomplished in the following way. We construct a directed graph G in the following way. We map each state to a distinct vertex in G. A one-rule event that uniquely triggers a state transition from a state u to a state v is mapped 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. ■

Theorem 5 Temporal projection regarding one-rule events, arbitrarily ordered, in 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 directed acyclic graphs to an instance of the temporal projection problem PRJ1. (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 cause 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 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 one-rule events e_1 and e_2 respectively, the imposed ordering constraint on the events e_1 and e_2 prevents arcs e and e' from appearing in the 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 prove the NP-Completeness of this special case. ■

Theorem 6 Temporal projection regarding general events, arbitrarily ordered, in a state space of size $O(1)$ is solvable in polynomial time.

Proof. For this special case, the task in the PRJ1 variant 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 the 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 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 bipartite match problem can be solved in polynomial time [PS82], we can solve this special case of temporal projection problem in polynomial time. ■

Theorem 7 Temporal projection regarding events partially ordered as $O(1)$ number of chains of length $O(|\mathcal{E}|)$ and $O(\log |\mathcal{E}|)$ number of chains of $O(1)$ length in a polynomial-size state space can be solved in polynomial time.

Proof. Given a problem instance $\langle \mathcal{P}, s_0, \mathcal{G}, \mathcal{E}, \mathcal{O} \rangle$, the number of vertices in the graph constructed by the global search algorithm in Section 3 is $|S_{\mathcal{P}}| \prod_{1 \leq i \leq m} (1 + l_i)$, where $|S_{\mathcal{P}}|$ is the size of the state space, m is the number of chains, and l_i is the number of events in the ith chain. If $l_i = O(|\mathcal{E}|)$ and m is a constant, or $l_i = O(1)$ and $c = O(\log |\mathcal{E}|)$, 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. ■

Theorem 8 The goal $\mathcal{G}$ is true after all events occur if and only if for each region R, the regional subgoal $\mathcal{G}_R$ of R is true immediately after the events in R have all occurred.

Proof. According to the definition, all the 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 9 The set of coupling conditions of a region R is composed of two disjoint subsets: the set 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 is dependent on events not in R while 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 region R', p is also an abstract condition of this child region R'. Suppose p is a subgoal condition of R, 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 regions of R. Therefore, if p is in the union of the set of abstract condition and the set of coupling condition of R, 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 regions of R. In other words, p is a subgoal condition of R. Therefore if p is a coupling condition of R, p is either an abstract condition of R or a subgoal condition of R. ■

Theorem 10 Procedure Localized-Reasoning is sound and complete.

Proof. By induction on the levels of the region hierarchy (from the bottom level up), we have the following two properties for procedure Region-Absrtaction by using Corollary 2. First, for each pair (s, f) in Δ_R derived in procedure Localized-Reasoning, (1) the incremental subgoal $\tilde{\mathcal{G}}_R$ is true in f, (2) there exists an event sequence in $\mathbf{Q}_{\mathcal{O}}^{R}$ whose trajectory starts in a state where s is true and ends in a state where f is true, and (3) the pair (s, f) is encoded as a causal rule $\alpha \rightarrow \beta$ where α (β) is the abstract state of $\mathcal{A}_R$ in s (f).
Second, the regional subgoal $\mathcal{G}_R$ of a region R can be achieved immediately after the events in R occur if and only if (1) there exists a pair (s, f) in Δ_R derived by procedure Region-Absrtaction, and (2) the abstract state s of the coupling conditions of R can be true immediate before the events in R occur.
Third, given (α, β) as an input where $\alpha \rightarrow \beta$ is a rule of the abstract event e_R, procedure Generate-Sequence can (1) determines a pair (s, f) in Δ_R which is encoded as a causal rule (α, β), and (2) generate an event sequence in $\mathbf{Q}_{\mathcal{O}}^{R}$ such that regional subgoals is true immediately after all events in R occur if α is true immediate before the events in R occur.
Note that (1) $(\langle\rangle \rightarrow \langle\rangle)$ is the only rule in the abstract event of the root region, since the root region has no abstract conditions, and (2) the empty expression $\langle\rangle$ is true immediately before any events occur. According to the three properties above, procedure Localized-Reasoning derives an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$that ends in a goal state if and only there exist one.
■

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.

Proof. In procedure Localized-Reasoning, procedure Region-Abstracion and procedure Generate-Sequenc is (recursively) called once in each region, both of which use procedure Temporal-Search for local searches in individual regions. In particular, in each region procedure Temporal-Search is called for $2^{|\mathcal{A}_R|}$ times by Step 2 of procedure Region-Abstracion. By Theorem 2, In a region R, procedure Temporal-Search runs in $O(2^{2(b_R+c_R)})$ time according to Theorem 2, 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. (1) In the 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 correspons to a region hierarchy with n events. (2) In each region, we also have

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

The whole thing is then simplified to the $O(n \times 8^{\mathcal{L}})$ time bound. ■

References

[BK91] C. Bäckström and I. Klein. Parallel non-binary planning in polynomial time. In *Proceedings IJCAI 12*, pages 268–273. IJCAII, 1991.

[Byl91] Tom Bylander. Complexity results for planning. In *Proceedings IJCAI 12*, pages 274–279. IJCAII, 1991.

[Cha87] David Chapman. Planning for conjunctive goals. *Artificial Intelligence*, 32:333–377, 1987.

[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.

[GJ79] Michael R. Garey and David S. Johnson. *Computers and Intractibility: 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.

[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.

[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.

[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.

[YT90] Qiang Yang and Josh D. Tenenberg. Abtweak: Abstracting a nonlinear, least commitment planner. In *Proceedings AAAI-90*, pages 204–209. AAAI, 1990.