Exploiting Locality in Temporal Reasoning

Shieu-Hong Lin and Thomas Dean

Department of Computer Science
Brown University
Providence, Rhode Island 02912

CS-93-24
June 1993

Exploiting Locality in Temporal Reasoning

Shieu-Hong Lin Thomas Dean[1]
Department of Computer Science
Brown University, Providence, RI 02912

Abstract

Temporal reasoning with uncertainty about the ordering of events is important in a wide variety of applications. Previous research shows that the associated decision problems are hard even for very restricted cases. In this paper, we investigate using the temporal locality of events and the spatial locality of state spaces to speed inference. We propose a new perspective for representing cause-and-effect relationships based on finite state automata. This perspective exposes the sources of complexity in temporal reasoning more clearly, and facilitates transforming rule-based temporal reasoning problems into graph reachability problems. We present both positive and negative results that provide insight into the complexity of temporal reasoning.

Content areas: Temporal Reasoning, Reasoning about Action

[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.

1 Introduction

Given a set of events, a set of ordering constraints on the events, and a description of the cause-and-effect relationships involving the associated event types, it is often useful to determine whether a given set of propositions can all be true after an event sequence. This is the essence of the temporal reasoning problem considered in this paper. 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 restrictions on the ordering of event instances. In special cases, temporal reasoning is harder than the corresponding planning problem [Nebel and Bäckström, 1991]. This happens because in these special cases, the type, the number, and the occurring order of events can be controlled by the planner. In most cases, as long as the introduced events are not under control, however, both problems become computationally equivalent. In particular, temporal reasoning with partially ordered event instances and cause-and-effect relationships described by STRIPS-like propositional operator schemas is hard [Dean and Boddy, 1988].

Previous complexity results on temporal reasoning and planning primarily address problems characterized in terms of rule-based cause-and-effect descriptions [Bäckström and Klein, 1991] [Bylander, 1991] [Chapman, 1987] [Dean and Boddy, 1988] [Gupta and Nau, 1991]. In this paper, we explore temporal reasoning from a different perspective. Events are considered in terms of input symbols in finite automata. The cause-and-effect relationships are considered in terms of state-transition relations in finite automata. We discover that the ordering constraints on events, the size of the state space, and the structure of the state transitions all contribute to the complexity of temporal reasoning. This new perspective suggests a different approach to solving temporal reasoning and planning problems by exploiting the temporal locality in event ordering and the spatial locality of state spaces.

In AI, the cause-and-effect relationships governing the consequences of actions are typically represented in terms of a set of rules. Each rule is associated with a set of antecedent conditions (or *preconditions*) and a set of consequent conditions (or *postconditions*). Alternatively, we can view the cause-and-effect relationships involving events as state-transition relations of finite automata. The conditions in rules define a state space where each condition corresponds to a binary state variable. Events are considered as input symbols. Rules

determine how events trigger state transitions in the state space. In general, we wish to allow arbitrary transition relations. From this perspective, the task in temporal reasoning is to determine whether a goal state can be reached, given a set of possible event sequences. The possible event sequences are determined by the ordering constraints on the events.

In Section 2, we formally define the temporal reasoning problem and contrast the two different perspectives. In Section 3, we show that the ordering constraints on events, the size of the state space, and the structure of state transitions all contribute to the complexity of temporal reasoning. Temporal reasoning turns out to be hard even if events are totally unordered and the involved state space is polynomial in the size of the problem instance. This result shows that it is necessary to exploit the structure of event ordering and state spaces. In Section 4, we explore the opportunities for using temporal locality and spatial locality to solve the temporal reasoning problem. We study temporal locality introduced by two types of ordering constraints. Event subsets are hierarchically encapsulated, or interleaved by these ordering constraints. A temporal reasoning algorithm is developed to hierarchically exploit the temporal locality introduced by these ordering constraints. We show that spatial locality may reduce the sizes of the state spaces involved in the hierarchical problem solving. Our techniques provide a possible way to reduce temporal reasoning to graph reachability. Our results do not depend on any severe restrictions on the causal rules. Finally, in Section 5, we discuss the possibility of applying the concept of planning by hierarchical abstraction spaces [Knoblock, 1991] to temporal reasoning, and point out that we cannot characterize abstraction hierarchies for temporal reasoning in the same way in planning, since we need to consider the ordering constraints on events and the limited number of available events.

2 Representations and Perspectives

2.1 Temporal Projection and Temporal Reachability

The *temporal projection problem* was studied by Dean and Boddy [Dean and Boddy, 1988]. The problem instance is defined by

- a set of event types, $\mathcal{T} = \{type_1, type_2, \ldots, type_n\}$;
- a set of conditions, $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$;
- a set of causal rules, $\mathcal{R} = \{r_1, r_2, \ldots, r_o\}$, of the form $\langle t, \varphi, \alpha, \delta \rangle$ composed of:
  - a triggering event type, $t \in \mathcal{T}$,
  - a set of antecedent conditions, $\varphi \subseteq \mathcal{P}$,
  - a set of added conditions, $\alpha \subseteq \mathcal{P}$,
  - a set of deleted conditions, $\delta \subseteq \mathcal{P}$;
- a set of actual events, $\mathcal{E}$, where $\forall e \in \mathcal{E}, type(e) \in \mathcal{T}$;
- a set of initial conditions, $\mathcal{I} \subseteq \mathcal{P}$;
- a partial order $\prec$ on $\mathcal{E}$.

The task in temporal projection is to determine whether a specified condition $p \in \mathcal{P}$ is true immediately following a specified event $e \in \mathcal{E}$ in some total order consistent with the partial order $\prec$. When an instance of a given event type occurs, the status, true or false, of a condition immediately following an instance is determined as follows. A condition is true immediately following an instance if and only if either

(i) the condition appears in the added conditions of a rule (associated with the event type) all of whose antecedent conditions are true immediately prior to the instance, *or*

(ii) the condition is true immediately prior to the instance *and* it is not the case that the condition appears in the deleted conditions of a rule (associated with the event type) all of whose antecedent conditions are true immediately prior to the instance.

Events in $\mathcal{E}$ occur as a sequence. A partial order provides an example of a set of ordering constraints. For example, given $e_i \prec e_j$, e_i must appear before e_j in any possible event sequence. Ordering constraints $\prec$ on $\mathcal{E}$ determine a set of possible event sequences. In this paper, we exploit the temporal locality introduced by ordering constraints on events. Other ordering constraints are investigated in Section 4. We define the following terms regarding event sequences and ordering constraints.

- An event sequence $\mathbf{q}$ (of length h) over a set of events $\mathcal{E}$ is a sequence $\langle e_1, e_2, \ldots, e_h \rangle$ where $e_i \in \mathcal{E}$, $1 \leq i \leq h \leq |\mathcal{E}|$, and $e_i \neq e_j$ if $i \neq j$. $\mathbf{q}' = \langle e_1, e_2, \ldots, e_{h'} \rangle$ is a subsequence of $\mathbf{q} = \langle e_1, e_2, \ldots, e_{h'}, \ldots, e_h \rangle$.
- Given a set of ordering constraints $\mathcal{O}$ on $\mathcal{E}$, $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ is composed of the event sequences over $\mathcal{E}$ which are (i) of length $|\mathcal{E}|$ and (ii) consistent with the ordering constraints in $\mathcal{O}$.
- $\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}} = \{\mathbf{q}' \mid \exists \mathbf{q} \in \mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ where $\mathbf{q}'$ is a subsequence of $\mathbf{q}\}$.

The partial order $\prec$ on $\mathcal{E}$ defines a set of possible event sequences $\mathbf{Q}_{\prec}^{\mathcal{E}}$.

In this paper, we focus on the following *temporal reachability problem*, which is closely related to the temporal projection problem. The problem instance is defined by $\langle \mathcal{T}, \mathcal{P}, \mathcal{R}, \mathcal{I}, \mathcal{G}, \mathcal{E}, \mathcal{O} \rangle$ where

- $\mathcal{T}$ is a set of event types,
- $\mathcal{P}$ is a set of conditions,
- $\mathcal{R}$ is a set of causal rules,
- $\mathcal{I}$ is a set of initial conditions where $\mathcal{I} \subseteq \mathcal{P}$,
- $\mathcal{G}$ is a set of goal conditions where $\mathcal{G} \subseteq \mathcal{P}$,
- $\mathcal{E}$ is a set of actual events,
- $\mathcal{O}$ is a set of ordering constraints on $\mathcal{E}$.

The set of ordering constraints $\mathcal{O}$ on $\mathcal{E}$ defines a set of event sequences $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$. The task in temporal reachability is to determine whether the conditions in $\mathcal{G}$ can all be true immediately following an event sequence $\mathbf{q}' \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$.

Lemma 1 shows that complexity results for the temporal reachability problem regarding partial orders on events can be applied to the temporal projection problem.

Lemma 1 *Temporal reachability regarding a partial order on events can be reduced to temporal projection in polynomial time.*

Proof. We first transform an instance of temporal reachability in the following way. A new condition p_0 and an event instance e_0 of a new event type t_0 are added. Event type t_0 is associated with a unique causal rule $r_0 = \langle t_0, \mathcal{G}, \{p_0\}, \mathcal{G}\rangle$. Event e_0 is allowed to occur after any event in $\mathcal{E}$. Temporal projection is then applied to determine whether p_0 can be true immediately following event e_0. □

2.2 A Perspective based on Finite Automata

An alternative perspective on cause-and-effect relationships involves representing them in terms of state transitions in a finite state space. Given an instance of the temporal reachability problem, $\mathcal{T}$, $\mathcal{P}$, $\mathcal{R}$, $\mathcal{I}$, and $\mathcal{G}$ determine a nondeterministic finite automaton $M = (S, \mathcal{T}, \Delta, s_0, F)$ in the following way.

- The set of conditions $\mathcal{P} = \{p_1, p_2, \ldots, p_m\}$ determines a state space S. A state is represented as a vector $u = (x_1, x_2, \ldots, x_m)$ in $\{0,1\}^m$. If condition p_i is true at state u, $x_i = 1$; otherwise $x_i = 0$.
- F denotes the set of states where the conditions in $\mathcal{G}$ are all true.
- The set of initial conditions $\mathcal{I}$ corresponds to an *initial state* $s_0 \in S$ where the conditions in $\mathcal{I}$ are true and the conditions in $\mathcal{P} - \mathcal{I}$ are false.
- We say that a rule $r = \langle t, \varphi, \alpha, \delta\rangle$ is *effective* at state u if all antecedent conditions in φ are true at state u. Event instances of type t may trigger state transitions at state u in the following ways:
  - transitioning from state u to any state v if there exists an effective rule r $= \langle t, \varphi, \alpha, \delta\rangle$ at state u where (i) the conditions in α are true at state v, (ii) the conditions in δ are false at state v, and (iii) the conditions in $\mathcal{P} - (\alpha \bigcup \delta)$ are true at v if and only if they are true at u;

– transitioning from state u to state u itself if there is no effective rule at state u associated with the event type t.

This provides a state transition relation Δ on $S \times \mathcal{T} \times S$.

Since events in $\mathcal{E}$ are symbols from the alphabet $\mathcal{T}$, event sequences in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ correspond to *input strings* to the finite automaton M. From this perspective, the following Lemma restates the task in temporal reachability based on the finite automata representation.

Lemma 2 *An instance of the temporal reachability problem determines a finite automaton $M = (S, \mathcal{T}, \Delta, s_0, F)$. The task in temporal reachability is to determine whether a state in F can be reached after an event sequence $\mathbf{q}' \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$.*

In other words, we want to determine whether there exists an input string $\mathbf{q}' \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$ accepted by M.

Similarly, the task in temporal projection can be restated based on the finite automata representation.

Lemma 3 *An instance of the temporal projection problem determines a finite automaton $M = (S, \mathcal{T}, \Delta, s_0, F)$. The task in temporal projection is to determine whether a state in F can be reached after an event sequence $\mathbf{q}' = \langle e_1, e_2, \ldots, e_h \rangle \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$, where $e_h = e$ is the specified event in the temporal projection problem.*

Proof. A finite automaton $M = (S, \mathcal{T}, \Delta, s_0, F)$ can be derived in the same way as in the case of the temporal reachability problem, except that in the latter F denotes the set of states where the specified condition p is true. □

When events are totally unordered and the state space involved is of polynomial size, the following lemma together with Lemma 1 show that temporal reachability and temporal projection are equivalent.

Lemma 4 *When there are no ordering constraints on events and the state space involved is of polynomial size, temporal projection can be reduced to temporal reachability in polynomial time.*

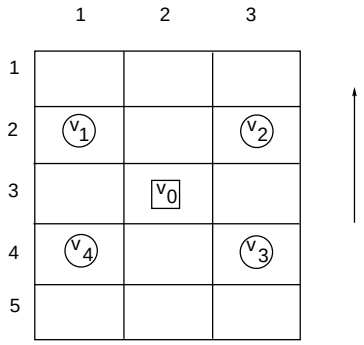

Figure 1: Highway traffic

Proof. From the perspective based on finite automata, we have the following reduction. We first specify the set of states as new final states where event e can trigger state transitions to final states. We then apply temporal reachability to determine whether we can reach any of the new final states, using events in $\mathcal{E} - \{e\}$. □

2.3 A Highway Traffic Example

Suppose we are driving a car on a three-lane highway at a constant speed. We can observe a nearby area around our car in a coordinate system centered on our car. The relative positions of vehicles in this area remain the same if all cars move at the same speed.

Figure 1 depicts the current relative positions of cars v_0, v_1, v_2, v_3, where v_0 is our car and the arrow indicates the moving direction of the cars. The observed area is divided into five rows and three columns where columns correspond to lanes. Our car v_0 is located at the third row of the second column, and its position remains fixed relative to the coordinate system. As soon as other cars speed up, slow down, or change lanes, the relative positions change.

The set of conditions $\mathcal{P}$ is composed of conditions in the form of $(At\ v_k\ i\ j)$ or $\overline{(At\ v_k\ i\ j)}$ indicating that car v_k is or is not currently at the ith row of the jth column, and conditions in the form of $(Occupied\ i\ j)$ or $\overline{(Occupied\ i\ j)}$ indicating that the ith row of the jth column is or is not occupied.

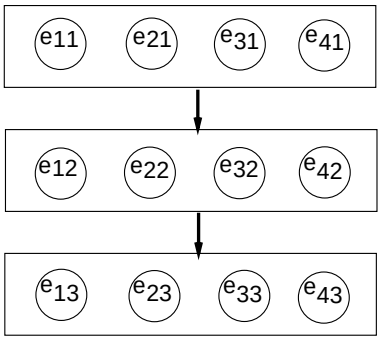

Figure 2: Uncertainty in event ordering

In this example, event e_{ij} models the action of car v_i in the jth time unit. For example, car v_1 wants to move to lane b in the first time unit. Car v_1 moves to the right directly if lane b is not occupied. Otherwise, v_1 slows down if no other cars closely follow it. Event e_{11} of type $type_1$ models the described behavior, which is associated with the following two rules:

- $r_1 = \langle type_1, \varphi_1, \alpha_1, \delta_1 \rangle$ where
  - $\varphi_1 = \{(At\ v_1\ 2\ 1), \overline{(Occupied\ 2\ 2)}\}$,
  - $\alpha_1 = \{(At\ v_1\ 2\ 2), (Occupied\ 2\ 2)\}$,
  - $\delta_1 = \{(At\ v_1\ 2\ 1), (Occupied\ 2\ 1)\}$, *and*
- $r_2 = \langle type_1, \varphi_2, \alpha_2, \delta_2 \rangle$ where
  - $\varphi_2 = \{(At\ v_1\ 2\ 1),(Occupied\ 2\ 2),\overline{(Occupied\ 3\ 1)}\}$,
  - $\alpha_2 = \{(At\ v_1\ 3\ 1), (Occupied\ 3\ 1)\}$,
  - $\delta_2 = \{(At\ v_1\ 2\ 1), (Occupied\ 2\ 1)\}$.

Suppose we want to model the behaviors in the coming three time units for the four cars around us. An event set $\mathcal{E}$ of twelve events is depicted in Figure 2, where event e_{ij} models the action taken by car v_i in the jth time unit. The following ordering constraints $\mathcal{O}$ on $\mathcal{E}$ determine the possible event sequences. In each time unit,

- four events occur as an atomic unit to model the interactions among the four cars, *and*
- these four events can occur in arbitrary order to model the uncertain interactions among the cars.

For example, if car v_2 also wants to move to lane b, the relative positions of cars after the first time unit will depend on which car takes the action first. If we wish to exit the highway soon, the situation depicted in Figure 3 is undesirable for we might be trapped and prevented from exiting the highway. We may infer the possibility of reaching that situation in the coming three time units. This corresponds to an instance of the temporal reachability problem where

- the set of conditions $\mathcal{P}$ represent the relative positions of the cars,
- the set of initial conditions $\mathcal{I}$ represents the initial relative positions of the cars,
- (*Occupied* 2 2), (*Occupied* 3 2), (*Occupied* 3 3), (*Occupied* 3 4) compose the set of goal conditions $\mathcal{G}$,
- the set of event types $\mathcal{T}$ is composed of event types like $type_1$, and the set of causal rules R is composed of causal rules like r_1, r_2,
- $\mathcal{E}$ is composed of the twelve events to model the actions of the cars,

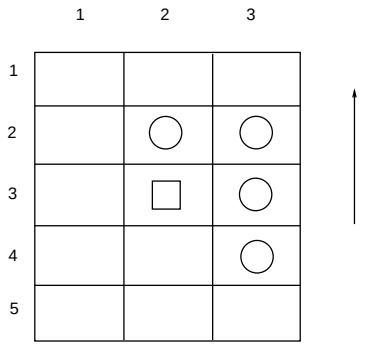

Figure 3: A undesirable situation to avoid

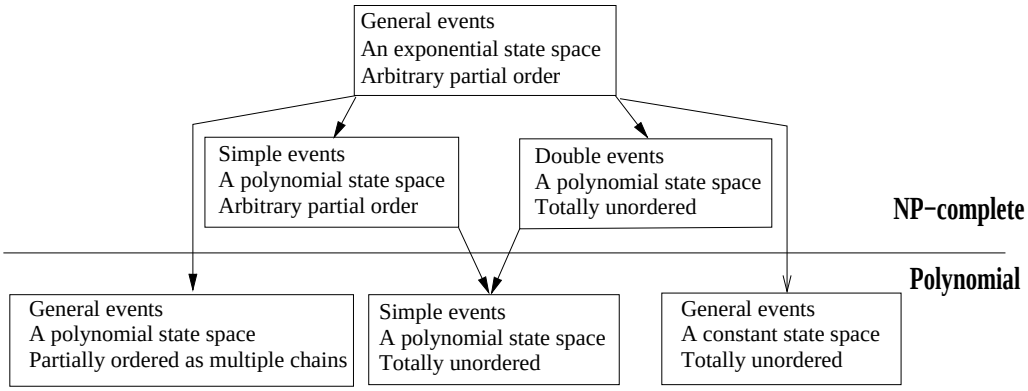

Figure 4: Complexity trade-offs for temporal reachability

- the ordering constraints in $\mathcal{O}$ determine the set of possible event sequences $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$.

3 Complexity Trade-offs

Previous research on the complexity of temporal projection has primarily focused on the structure of causal rules [Dean and Boddy, 1988] [Nebel and Bäckström, 1991]. Causal rules determine how event instances trigger state transitions. In this paper, we find that the constraints on event ordering and the size of the involved state space also play important roles. Figure 4 depicts the complexity trade-offs of temporal reachability (and thus temporal projection too) involving these three factors. Ordering constraints on events are classified as one of (i) arbitrary partial order, (ii) totally unordered (no ordering constraints), or (iii) *partially ordered as multiple chains*. We say that events are *partially ordered as multiple chains* [2] if (1) the events are partitioned into a constant number of disjoint event subsets, and (2) the events in each subset are totally ordered (forming a chain).

State space size is specified as one of constant, polynomial, or exponential, in the size of the problem instance $(\mathcal{T}, \mathcal{P}, \mathcal{R}, \mathcal{I}, \mathcal{G}, \mathcal{E}, \mathcal{O})$. Events are categorized as *simple*, *double* or general according to how they trigger state transitions. The state transitions are deter-

[2] We say that events are *partially ordered as chains* when the events are partitioned into arbitrary number of disjoint subsets and the events in each event subset are totally ordered.

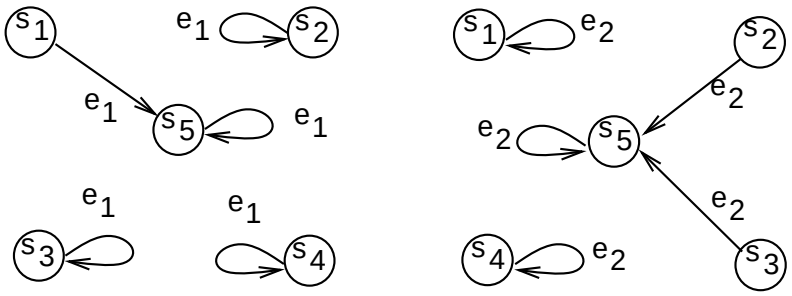

Figure 5: State transitions triggered by events

ministic for both simple events and double events. Simple events map exactly one state to some other state and all others to themselves. Double events map no more than two states to states other than themselves. General events allow nondeterministic state transitions. In Figure 5, five states are involved. Event e_1 is a simple event, and event e_2 is a double event.

In the following, we provide proofs and algorithms to justify the claims implicit in Figure 4. These proofs and algorithms are all based on the finite automata representation. All of the problems are clearly in NP. We first define the following terms.

- An event sequence **q** is *coherent from state s to state t* if (i) the event ordering in **q** is consistent with the ordering constraints on the events, (ii) the state transitions are deterministic for all events in **q**, (iii) starting from state s each event maps a state to a distinct state, and (iv) state t is reached after the last event.
- The *trajectory* of a coherent event sequence **q** is the sequence of states visited when the events in the event sequence occur.

Theorem 1 *Temporal reachability regarding double 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 [Garey and Johnson, 1979] 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. An instance of the disjoint forbidden pairs of arcs problem can be reduced to an instance of the temporal reachability problem where there is only a final state. Given an instance of disjoint forbidden pairs, the nodes in the directed graph are considered as states. Each directed edge (u, v) in the graph corresponds to state transition from u to v triggered by some event. We map each forbidden pair of arcs to a distinct double event, and each of the remaining arcs to a simple event. The initial node and the goal node are mapped to the initial state s_0 and the final state f respectively. If we can find a coherent event sequence from the initial state s_0 to the final state f, then the answer to the forbidden pairs problem is yes. Since an event corresponds to a forbidden pair of edges, it can only map one of its two possible state transitions. The trajectory of the coherent event sequence then corresponds to a feasible path from s_0 to f. Similarly, a path from s_0 to f without using both arcs in any forbidden pair corresponds to a coherent event sequence from s_0 to f. □

This result implies that we have very limited ability to deal with events that occur without temporal locality, even in a state space of polynomial size.

Theorem 2 *Temporal reachability regarding simple events, totally unordered, in a polynomial-size state space, or regarding general events, totally unordered, in a constant-size state space are both solvable in polynomial time.*

Proof. States are considered as nodes. State transitions triggered by events are considered as directed edges. This describes a graph. A simple event then corresponds to a directed edge in the graph. Temporal reachability in the former case can be directly reduced to graph reachability, since a coherent event sequence from the initial state s_0 to a final state f corresponds to a path from s_0 to f. In the second case, there is only a constant number of directed paths from the initial state s_0 to a final state f. For each directed path, we determine whether there is an event sequence which can trigger state transitions along such a trajectory path. The answer is yes if we can match distinct events to the directed edges in the trajectory path. Note that an event can match a directed edge (u, v) if the event can trigger a state transition from state u to state v. This problem is then reduced to the bipartite matching problem. □ *This result implies that we can deal with smaller state spaces and simpler state transition rules better.*

Theorem 3 *Temporal reachability regarding simple events with an arbitrary partial order on events, in a polynomial-size state space is NP-Complete.*

Proof. The forbidden pairs of arcs problem on directed acyclic graphs is NP-Complete [Garey and Johnson, 1979]. By adapting the NP-Completeness proof for simple event systems in [Nebel and Bäckström, 1991], an instance of the forbidden pairs of arcs problem on a directed acyclic graph can be reduced to an instance of the temporal reachability problem where there is only a final state. The nodes in the given directed acyclic graph are considered as states. Each arc is mapped to a unique simple event. For each forbidden pair of arcs (e_1, e_2) assign precedence $e_1 \prec e_2$ if there is a directed path with arc e_2 preceding arc e_1. Otherwise, assign precedence $e_2 \prec e_1$ if there is a directed path with arc e_1 preceding arc e_2. Since the graph is acyclic, the precedence relation among events is a partial order. The trajectory of a coherent event sequence from the initial state s_0 to the final state f never contains both arcs of a forbidden pair. Such a trajectory corresponds to a feasible path from s_0 to f. Similarly, edges in a feasible path from s_0 to f correspond to a coherent event sequence from s_0 to f. □

The temporal reachability problem regarding events partially ordered as chains can be solved using the following procedure. Temporal reachability is reduced to graph reachability in this case. This procedure is also used in Section 4 for hierarchically solving the temporal reachability problem.

Procedure MultiChain$(\mathcal{E}, \mathcal{O}, I)$
input: (1) $\mathcal{E}$, a set of events; (2) $\mathcal{O}$, a set of ordering constraints on $\mathcal{E}$ where events are partially ordered as chains; (3) I, a set of initial states.
output: (1) the set of states reached after the event sequences in $\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$; (2) the set of states reached after the event sequences in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$; (3) the sets of states that we may reach immediately before a specific event e occur, $\forall e \in \mathcal{E}$.

- Suppose events are partially ordered as c chains with l_i events in the ith chain. Construct the following directed graph $G = (V, E)$.
  - Each state $s \in S$ is expanded into $\prod_{1 \leq i \leq c}(1 + l_i)$ nodes of the form $(s, x_1, x_2, \ldots, x_c)$ where $0 \leq x_i \leq l_i$. These nodes comprise the vertex set V of G.
  - Node $(s, x_1, x_2, \ldots, x_c)$ indicates that we are at state s when $\forall i$ the first x_i events in the ith chain have occurred.

- Construct a directed edge from $(s, x_1, x_2, \ldots, x_k - 1, \ldots, x_c)$ to $(t, x_1, x_2, \ldots, x_k, \ldots, x_c)$ if the x_kth event in the kth chain can trigger a state transition from s to t. These edges comprise the edge set V of G.

- We apply the standard graph reachability algorithms to derive the sets of states we want.
  - State t is reachable from state s_0 ($s_0 \in I$) after an event sequence in $\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$ if and only if $\exists(t, x_1, x_2, \ldots, x_c)$ is reachable from $(s_0, 0, 0, \ldots, 0)$ in G.
  - State t is reachable from state s_0 ($s_0 \in I$) after an event sequence in $\mathbf{Q}_{\mathcal{O}}^{\mathcal{E}}$ if and only if $(t, l_1, l_2, \ldots, l_c)$ is reachable from $(s_0, 0, 0, \ldots, 0)$ in G.
  - Suppose e is the rth event in the kth chain. Immediately before event e occurs, we can reach state t from state s_0 ($s_0 \in I$) if and only if $\exists(t, x_1, x_2, \ldots, x_k = r - 1, \ldots, x_c)$ is reachable from $(s_0, 0, 0, \ldots, 0)$ in G.

Theorem 4 *Temporal reachability regarding general events, partially ordered as chains, in a polynomial-size state space is solvable in polynomial time if the product (over all chains) of 1 plus the chain length is of polynomial size.*

Proof. Suppose the events in $\mathcal{E}$ are partially ordered as multiple chains by the ordering constraints in $\mathcal{O}$. We apply **MultiChain**$(\mathcal{E}, \mathcal{O}, \{s_0\})$ to determine the set of states reachable after the event sequences in $\overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$. The number of nodes in the graph constructed by procedure **MultiChain** is equal to $|S| \prod_{1 \le i \le c}(1 + l_i)$ where $|S|$ is the size of the state space and l_i is the number of events in the ith chain. In this case, the graph G is of polynomial size if the product (over all chains) of 1 plus the chain length ($\prod_{1 \le i \le c}(1 + l_i)$) is of polynomial size. Graph reachability and thus temporal reachability is solvable in polynomial time. □

Corollary 1 *Temporal reachability regarding general events, partially ordered as chains, in a polynomial-size state space can be solved in polynomial time for a constant number of chains of length $O(n)$ or $O(\log n)$ number of chains of constant length, where n is the size of the problem instance.*

4 Exploiting Locality

4.1 Temporal Locality in Event Ordering

As shown in Section 3, temporal reachability regarding general events is hard if the events are totally unordered and the state space is polynomial in the size of the problem instance. When totally unordered, events can occur in arbitrary order. However, there is temporal locality in event ordering if the occurrences of events or subsets of events closely relate to one another. For example, a subset of events may always occur as an atomic unit; all the other events either occur before or after the subset of events. Similarly, several atomic units may always occur as an atomic unit. In this way, temporal locality can be introduced at many levels. Suppose we have a set of events $\mathcal{E}$. The following *Encapsulate* type of ordering constraints can introduce temporal locality in an event subset $X = \{e_1, e_2, \ldots, e_k\}$ in $\mathcal{E}$.

- *Encapsulate*(X): Events in X are required to occur as an atomic unit. All the other events either occur before or after the subset of events.

We say that X is a *local event subset* in $\mathcal{E}$ if X is constrained by the *Encapsulate* type of ordering constraints. The event set $\mathcal{E}$ and each event in $\mathcal{E}$ are local event subsets in $\mathcal{E}$, since they are implicitly constrained by the *Encapsulate* type of ordering constraints. For any two local event subsets X and Y in $\mathcal{E}$, we require either that X and Y are disjoint or that one is a proper subset of the other. Suppose X is a subset of Y. We say that X is *maximal* in Y if X is not a proper subset of any other local event subset in Y. In this way, the local event subsets in $\mathcal{E}$ form a hierarchy.

Given a local event subset Y, the following *Interleave* type of ordering constraints can introduce a partial order on the maximal local event subsets in Y. These event subsets are partially ordered as chains.

- *Interleave*$(\langle X_1^1, X_1^2, \ldots\rangle, \langle X_2^1, X_2^2, \ldots\rangle, \ldots, \langle X_c^1, X_c^2, \ldots\rangle)$: X_i^j's are the maximal local event subsets in Y. $\forall i$, $X_i^j \prec X_i^k$ if $j < k$. $\langle X_i^1, X_i^2, \ldots\rangle$ forms a chain of local event subsets. The events in X_i^j must occur before the events in X_i^k if $j < k$.

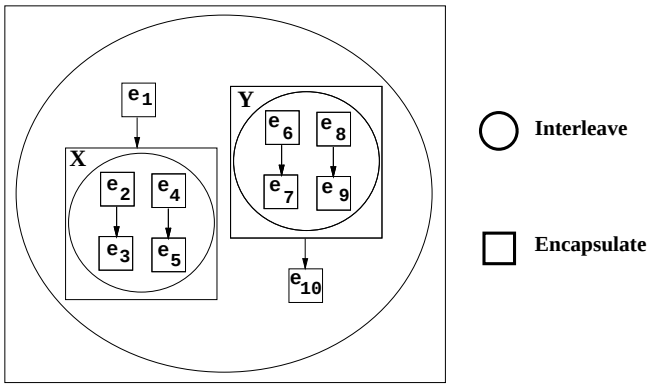

Figure 6: Temporal locality in local event subsets

4.1.1 An Example

Figure 6 shows the ordering constraints on ten events where $X = \{e_2, e_3, e_4, e_5\}$ and $Y = \{e_6, e_7, e_8, e_9\}$:

- $Encapsulate(X)$,
- $Interleave(\langle\{e_2\}, \{e_3\}\rangle, \langle\{e_4\}, \{e_5\}\rangle)$,
- $Encapsulate(Y)$,
- $Interleave(\langle\{e_6\}, \{e_7\}\rangle, \langle\{e_8\}, \{e_9\}\rangle)$,
- $Encapsulate(\{e_1\} \bigcup X \bigcup Y \bigcup \{e_{10}\})$.
- $Interleave(\langle\{e_1\}, X\rangle, \langle Y, \{e_{10}\}\rangle)$.

$X, Y, \{e_1\}, \{e_{10}\}$ are the maximal local event subsets in the event set composed of these ten events. The events in X always occur together. In X, e_3 never occurs before e_2 and e_5 never occurs before e_4. The events in Y always occur together. In Y, e_6 always occurs before e_7. e_8 always occurs before e_9. e_1 always occurs before the events in X. e_{10} always occurs after the events in Y.

4.2 Hierarchical Problem Solving

Given a local event subset X and a set of ordering constraints $\mathcal{O}$ on X, X can be abstracted as an *abstract event* e_X. Suppose that we can reach state v from state u after an event sequence $\mathbf{q} \in \mathbf{Q}_{\mathcal{O}}^{X}$. Then the abstract event e_X may trigger a state transition from u to v. This describes a state-transition relation involving e_X, since we may reach another state v' from state u after another event sequence $\mathbf{q}' \in \mathbf{Q}_{\mathcal{O}}^{X}$. In other words, e_X summarizes the possible state transitions caused by the events in X as a whole.

When the ordering of events is governed by the *Encapsulate*, and *Interleave* types of ordering constraints, the temporal reachability problem can be solved hierarchically. In the hierarchical problem solving, local event subsets are abstracted as abstract events. For example, temporal reachability regarding events in Figure 6 can be solved in two levels. At the top level, we determine the possibility of reaching any goal state, given (abstract) events e_1, e_{10}, e_X, e_Y. If the answer is negative, we first determine the possible states that we may reach immediately before the events in X (Y) occur. Then at the second level, we determine the possibility of reaching any goal state, given the event subset X (Y) and the possible states that we may reach immediately before the events in X (Y) occur.

In the following, the procedure **MultiChain** in Section 3 is used as a subroutine in hierarchical problem solving. The following procedure recursively derives the abstract events for the local event subsets in a set of events.

Procedure Event-Abstraction($\mathcal{E}, \mathcal{O}$)
input: (1) $\mathcal{E}$, a set of events; (2) $\mathcal{O}$, a set of ordering constraints on $\mathcal{E}$.
output: the abstract events for the local event subsets in $\mathcal{E}$.

- For each maximal local event subset X in $\mathcal{E}$, we call **Event-Abstraction**($X, \mathcal{O}_X$) to derive the abstract events for the local event subsets in X where $\mathcal{O}_X$ is composed of the ordering constraints on X, $\mathcal{O}_X \subset \mathcal{O}$.
- Let $\mathcal{A}$ be composed of the abstract events for the maximal local event subsets in $\mathcal{E}$. Let $\prec$ be the partial order on $\mathcal{A}$ introduced by the *Interleave* ordering constraint on the maximal local event subsets in $\mathcal{E}$. We derive the abstract event $e_{\mathcal{E}}$ for $\mathcal{E}$ as follows.
  - For each state s, the abstract event $e_{\mathcal{E}}$ can trigger a state transition from state s to any state reachable after the event sequences in $\mathbf{Q}_{\prec}^{\mathcal{A}}$. We call **MultiChain**($\mathcal{A}, \prec, \{s\}$)

to determine this state-transition relation.

The following procedure recursively decomposes the temporal reachability problem and then solves it level by level.

Procedure Problem-Decomposition$(\mathcal{E}, \mathcal{O}, I, G)$

input: (1) $\mathcal{E}$, a set of events; (2) $\mathcal{O}$, a set of ordering constraints on $\mathcal{E}$; (3) I, a set of initial states; (4) G, a set of goal states.

output: yes if $\exists s \in I, \exists t \in G, \exists \mathbf{q} \in \overline{\mathbf{Q}}_{\mathcal{O}}^{\mathcal{E}}$ such that s is reachable from t after the event sequence $\mathbf{q}$; no, otherwise.

- We call **Event-Abstraction**$(\mathcal{E}, \mathcal{O})$ to derive the abstract events for all the local event subsets in $\mathcal{E}$.
- Let $\mathcal{A}$ be composed of the abstract events for the maximal local event subsets in $\mathcal{E}$. Let $\prec$ be the partial order on $\mathcal{A}$ introduced by the *Interleave* ordering constraint on the maximal local event subsets in $\mathcal{E}$. At the top level, we call **MultiChain**$(\mathcal{A}, \prec, I)$ to determine the set of states reachable after the event sequences in $\overline{\mathbf{Q}}_{\prec}^{\mathcal{A}}$.
- If we can reach states in G at this level, the answer to the problem is yes. Otherwise, we call **MultiChain**$(\mathcal{A}, \prec, I)$ to determine I_X, the set of states that we may reach immediately before e_X occurs $(\forall e_X \in \mathcal{A})$. I_X corresponds to the states that we may reach immediately before any event in X occurs.
- At the next level, we call **Problem-Decomposition**$(X, \mathcal{O}_X, I_X, G)$ for each maximal local event subset X in $\mathcal{E}$ where $\mathcal{O}_X$ is composed of the ordering constraints on X, $\mathcal{O}_X \subset \mathcal{O}$.

4.3 State Space Reduction in Temporal Reachability

Spatial locality in local event subsets: In the worst case, the procedure **Problem-Decomposition** deals with state spaces of size $2^{|\mathcal{P}|}$. $\mathcal{P}$ is the set of conditions. However, spatial locality in local event subsets may reduce the sizes of the state spaces involved in hierarchical problem solving. We say that event e involves condition p if p appears in a causal rule associated with the event type of e. In Figure 7, eleven conditions are involved. Event e_1 involves the conditions a, b, c, j. The events in $X = \{e_2, e_3, e_4, e_5\}$ involve the conditions a, b, d, e, f. The events in $Y = \{e_6, e_7, e_8, e_9\}$ involve the conditions a, b, g, h, i.

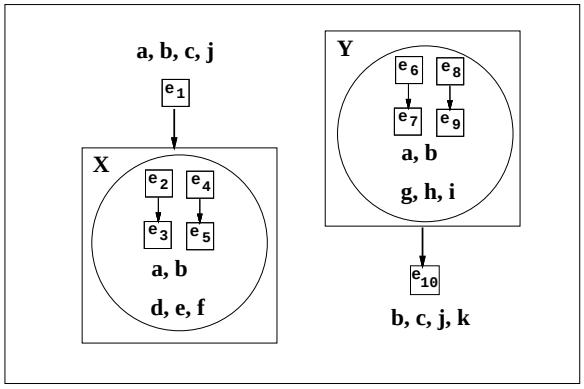

Figure 7: Spatial locality in local event subsets

Event e_{10} involves the conditions b, c, j, k. We want to determine whether we can reach any state where the conditions a, b, c are all true. Note that the conditions d, e, f (g, h, i) are only involved locally in event subset X (Y). It turns out that we only need to deal with state spaces involving no more than five conditions instead of all the eleven conditions in this problem instance.

- A state space involving the conditions a, b, d, e, f (a, b, g, h, i) is considered for deriving the abstract event e_X (e_Y) for the local event subset X (Y).
- At the top level, we deal with the events e_1, e_{10}, e_X, e_Y. The conditions d, e, f (g, h, i) are only locally involved in X (Y). So we only need to consider the possible status transitions of the conditions a, b triggered by e_X (e_Y). Since the events e_1, e_{10} involve the conditions a, b, c, j, k, a state space involving the conditions a, b, c, j, k is considered at the top level.
- At the top level, if we can reach a state where the conditions a, b, c are all true, the answer to the problem is yes. Otherwise the condition c must be made true at the top level, and the conditions a, b must be made true at the second level involving the event subsets X, Y. This is because the status of condition c cannot be changed by the events in X, Y.

- At the second level, a state space involving the conditions a, b, d, e, f (a, b, g, h, i) is considered for the event subset X (Y). The task is to determine whether we can reach a state where the conditions a, b are true, given events in X (Y).

- The possible statuses of the conditions a, b, d, e, f (a, b, g, h, i) immediately before the events in X (Y) occur can be determined as follows. We derive the set of states that we may reach at the top level immediately before e_X (e_Y) occurs. We then find out the possible statuses of the conditions a, b at these states, given that the condition c must be true at these states. The statuses of the conditions d, e, f (g, h, i) are the same as their initial statuses, since they are only locally involved in event subset X (Y).

Estimating the reachable state subspace: In some applications, there are bounds on the number of conditions being true at a time. For example, suppose the number of the added conditions is always less than or equal to the number of the deleted conditions in any causal rule. If there are m conditions initially true, then no more than m conditions can be true at a time. In this case, the size of the reachable state subspace is no more than $\sum_{1 \leq i \leq m} \binom{|\mathcal{P}|}{i}$ where $\mathcal{P}$ is the set of conditions.

Minimizing the finite automaton: The size of the state space can be reduced by replacing the finite automaton determined by the problem instance with an equivalent finite automaton involving fewer states. The algorithms for minimizing finite automata can be used to reduce the size of the state space.

5 Temporal Reasoning by Abstraction

Knoblock [Knoblock, 1991] discusses planning using abstraction hierarchies to reduce the size of the search space. Given an abstraction hierarchy for planning, variables are partitioned into levels. An abstract plan is searched at the highest level, and then refined level by level to incrementally achieve the goal conditions. In solving the temporal reachability problem, we determine whether or not there exists a possible event sequence to reach the

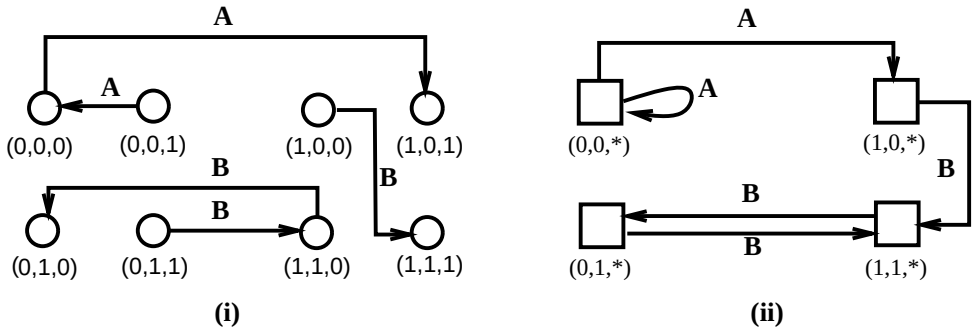

Figure 8: State transitions in an abstract state space

goal state subspace. The state trajectory of an event sequence is the set of states we may reach when the events in the event sequence trigger state transitions. We say that an event sequence can reach a state if the state is in the trajectory of the event sequence. In a possible event sequence, the ordering of the events must be consistent with the ordering constraints, and any event can occur at most once. Naturally, we can have an analogous concept of temporal reasoning by abstraction. We incrementally verify whether there is a possible event sequence to reach the goal state subspace at different *abstraction levels*.

At an abstraction level, an *abstract state space* spanned by a subset of the state variables is considered. These state variables are called the *base variables* at the abstraction level. States with the same configuration for the base variables are abstracted into an *abstract state*. The original state space is then abstracted as an abstract state space. The state transitions in the original state space are abstracted into the state transitions in the abstract state space. Figure 8-(i) shows the state transitions involving the events A, B in a state space spanned by three variables. Figure 8-(ii) shows the abstraction of the state transitions involving the events A, B in a state space spanned by two of the three variables. The states (1,1,1) and (1,1,0) in the original state space is mapped into an abstract state (1,1,*) in the abstract state space in this example. We say that (1,1,*) is the *abstraction* of (1,1,1) and (1,1,0) at an abstraction level involving the two variables.

We define an *abstraction hierarchy* as a sequence of abstraction levels where the set of base variables at a level is properly contained by the sets of base variables at lower levels. Given an abstraction hierarchy, the abstract state spaces can be spanned incrementally.

We try to incrementally refine a possible event sequence at a level to that at a lower level to reach the abstraction of the goal state subspace. If none of the abstraction of the goal state subspace can be reached by any possible event sequence at an abstraction level, obviously the goal subspace can never be reached at all. Otherwise, we can look at a lower abstraction level with a larger set of base variables and repeat this process again. However, backtracking between levels is needed to deal with all of the possibilities of reaching the goal state subspace.

To eliminate backtracking, there must be additional properties among abstraction levels. A *refinable abstraction hierarchy* is an abstraction hierarchy with the following property.

Definition 1 *For any abstract state t at level k, suppose t' is the abstraction of t at level k', $k' < k$. If there does exist a possible event sequence reaching t at level k, then every possible event sequence reaching t' at level k' can be refined to a possible event sequence reaching t at level k.*

Given a refinable abstraction hierarchy, there is no backtracking between abstraction levels when we are finding an event sequence to reach the goal state subspace. As soon as we fail to refine an event sequence at an abstraction level, we know that there exists no event sequence reaching the abstraction of the goal state subspace at this level. We do not need to backtrack to a higher level to explore other possibilities.

Our framework of temporal reasoning by abstraction is analogous to Knoblock's abstraction hierarchies for planning. However, Knoblock's results [Knoblock, 1991], characterizing abstraction hierarchies for planning, cannot be directly applied to temporal reasoning, since the ordering constraints on events and the limitation on the number of available events are not considered in abstraction hierarchies for planning. Characterizing refinable abstraction hierarchies or other kinds of useful abstraction hierarchies for temporal reasoning remains an interesting future research topic.

6 Conclusion

In this paper, we describe an alternative perspective on temporal reasoning, representing the cause-and-effect relationships as finite state automata. This perspective suggests and

we demonstrate theoretically that the ordering constraints on events, the size of the state space, and the state-transition rules all contribute to the problem complexity. It is shown that we have very limited ability to deal with general events if there is no temporal or spatial locality at all. We characterize the temporal locality induced by interleaving, encapsulating, and preempting subsets of events, and provide an algorithm that takes advantage of the locality in events in performing temporal reasoning. We also discuss temporal reasoning by abstraction, and point out the difficulty of directly applying the results of planning by abstraction to temporal reasoning.

References

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

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

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

[Dean and Boddy, 1988] Dean, Thomas and Boddy, Mark 1988. Reasoning about partially ordered events. *Artificial Intelligence* 36(3):375–399.

[Garey and Johnson, 1979] Garey, Michael R. and Johnson, David S. 1979. *Computers and Intractibility: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York.

[Gupta and Nau, 1991] Gupta, Naresh and Nau, Dana S. 1991. Complexity results for blocks-world planning. In *Proceedings AAAI-91*. AAAI. 629–633.

[Knoblock, 1991] Knoblock, Craig A. 1991. Search reduction in hierarchical problem solving. In *Proceedings AAAI-91*. AAAI. 686–691.

[Nebel and Bäckström, 1991] Nebel, Bernhard and Bäckström, Christer 1991. On the computational complexity of temporal projection and some related problems.