Available online at www.sciencedirect.com
ScienceDirect
Systems & Control Letters 56 (2007) 656–661
SYSTEMS & CONTROL LETTERS
www.elsevier.com/locate/sysconle

An algorithm for calculating indistinguishable states and clusters in finite-state automata with partially observable transitions☆

Weilin Wang[a], Stéphane Lafortune[a,*], Feng Lin[b]

[a]Department of EECS, University of Michigan-Ann Arbor, USA
[b]Department of ECE, Wayne State University, USA

Received 7 August 2006; accepted 17 March 2007
Available online 22 May 2007

Abstract

This paper presents a new algorithm for efficiently calculating pairs of indistinguishable states in finite-state automata with partially observable transitions. The need to obtain pairs of indistinguishable states occurs in several classes of problems related to control under partial observation, diagnosis, or distributed control with communication for discrete event systems. The algorithm obtains all indistinguishable state pairs in polynomial time in the number of states and events in the system. Another feature of the algorithm is the grouping of states into clusters and the identification of indistinguishable cluster pairs. Clusters can be employed to solve control problems for partially observed systems.

Keywords: Discrete event systems; Automata; Observability; State clusters; Supervisory control

1. Introduction

Consider a discrete event system modeled by a deterministic finite-state automaton G where some of the transitions of G are not observable. A pair of states (x_1, x_2) of G is said to be *indistinguishable* if G can generate two traces of events, s_1 and s_2, where s_i takes the system to x_i, $i=1, 2$, and such that s_1 and s_2 have the same observable projection. The need to identify all state pairs that are indistinguishable occurs in several classes of problems for partially observed discrete-event systems, most importantly in control [4], diagnosis [6], and communication [5]. Indistinguishable state pairs can be identified by examining the states of the *observer* of G, which is the deterministic automaton that is obtained after replacing every unobservable transition in G by the empty trace symbol and determinizing the result (see, e.g., [1]). However, the construction of the observer is worst-case exponential in the state space of G. To avoid this step, researchers have developed specific tests of polynomial complexity in the state space of G for several important properties thatarise in partially observed discrete-event system theory, such as observability [8] and diagnosability [10].

The contribution of this paper is to present a new algorithm that calculates indistinguishable state pairs and indistinguishable "cluster pairs" in polynomial complexity in the number of states and events in the system G. *Clusters* are sets of states characterized by their reachability with unobservable transitions from a given state, referred to as the *cluster head*. (A precise definition is given in the next section.) Once the pairs of indistinguishable states (clusters) have been identified, it is generally straightforward to test several important discrete-event system-theoretic properties such as observability [3], diagnosability [6], detectability [7], and feasibility (of communication policies) [5,9]. In fact, the algorithm herein was originally developed to test feasibility in the context of solving minimum communication problems in [9]. The realization that it is applicable in a variety of problems for partially observed discrete-event systems has motivated its separate presentation in this paper. Moreover, the notion of clusters, which is at the core of the algorithm, is of independent interest and can be used for solving control problems under partial observation, as is briefly sketched in Section 5 of this paper.

☆ The research of the first two authors is supported in part by ONR Grant N0001-14-03-1-0232 and by NSF Grant CCR-0325571. The research of the third author is supported in part by NSF Grant INT-0213651.
* Corresponding author.
E-mail address: stephane@eecs.umich.edu (S. Lafortune).

0167-6911/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.sysconle.2007.03.006

The next section presents the necessary notions and defines clusters. Section 3 is concerned with the relations that capture indistinguishable cluster pairs (relation $\mathscr{T}$) and state pairs (relation T). The algorithm is presented in Section 4 and examples of its application for (i) testing observability and (ii) feedback control based on clusters, are given in Section 5. A brief conclusion follows in Section 6.

2. Automata with partially-observed transitions

We model the discrete event system as a deterministic finite-state automaton

$$G = (X, E, f, \Gamma, x_0), \quad (1)$$

where X is the finite set of states, E is the finite set of events, $f : X \times E \to X$ is the partial transition function where $f(x, e) = y$ means that there is a transition labeled by event e from state x to state y, $\Gamma : X \to E$ is the active event function and $\Gamma(x)$ denotes the set of events e for which $f(x, e)$ is defined, and x_0 is initial state. We will often make a slight abuse of notation and write $(x', e, x) \in f$ whenever $f(x', e) = x$. We use $\mathscr{L}(G)$ to denote the language generated by G. For all $s \in \mathscr{L}(G)$ and $e \in E$, the transition function f for trace $se \in \mathscr{L}(G)$ is defined recursively as $f(x_0, se) = f(f(x_0, s), e)$ whenever $f(x_0, s)$ is defined. We assume that G is accessible, i.e., $\forall x \in X, \exists s \in \mathscr{L}(G), f(x_0, s) = x$.

To specify which transitions are observable to a controller or agent, we denote the set of transitions $\mathrm{TR}(G)$ of G as follows:

$$\mathrm{TR}(G) := \{(x, e) \in X \times E : e \in \Gamma(x)\}. \quad (2)$$

Some of these transitions are observable and some are not. This is specified by an index function

$$\mathscr{I} : \mathrm{TR}(G) \to \{0, 1\}, \quad (3)$$

where $\mathscr{I}(x, e) = 1$ means that the transition (x, e) is observable and $\mathscr{I}(x, e) = 0$ means that it is not. In classical discrete event control problems, the event set is partitioned as $E = E_o \cup E_{uo}$, where E_o is the set of observable events and E_{uo} is the set of unobservable events. This is a specific case of our definition for observation on transitions, which corresponds to

$$\mathscr{I}(x, e) := \begin{cases} 1 & \text{if } e \in E_o, \\ 0 & \text{if } e \in E_{uo}. \end{cases}$$

To reflect the fact that some transitions in G are not observable, we define the information mapping (or projection) $\theta : \mathscr{L}(G) \to E^*$ as follows:

$\theta(\varepsilon) := \varepsilon$, where ε denotes the empty trace;

$$\theta(se) := \begin{cases} \theta(s)e & \text{if } \mathscr{I}(f(x_0, s), e) = 1, \\ \theta(s) & \text{if } \mathscr{I}(f(x_0, s), e) = 0. \end{cases}$$

Therefore, if a trace s occurs in G, the controller or agent will see $\theta(s)$. For any sublanguage $L \subseteq \mathscr{L}(G)$, we define its θ – Projection as

$$\theta(L) := \{\theta(s) : s \in L\}.$$

In particular, $\theta(\mathscr{L}(G))$ describes the set of all possible observed traces for system G.

To construct an automaton for $\theta(\mathscr{L}(G))$, we first replace all unobservable transitions by ε and then convert the resulting nondeterministic automaton with ε transitions into a deterministic automaton using standard methods (see, e.g., [1]). We will call this automaton the "observer" and denote it by G_{obs}. To formally define G_{obs}, we first define the "unobservable reach" of a set of states $X' \subseteq X$, denoted by $UR(X')$, as the set of states in X that are reachable from X' via some unobservable transitions. G_{obs} is then given as

$$G_{obs} = (Q, E, h, q_0) := \mathrm{Ac}(2^X, E, h, q_0),$$

where Ac denotes the accessible part of an automaton; $q_0 := \mathrm{UR}(\{x_0\})$; and $h(q, e) := UR(\{x \in X : (\exists x' \in q) f(x', e) = x \wedge \mathscr{I}(x', e) = 1\})$. We emphasize that Q is the set of all accessible states, each of which is a subset of X. It is not difficult to show that $\mathscr{L}(G_{obs}) = \theta(\mathscr{L}(G))$.

Based on G_{obs}, we can define the relation $\Pi \subseteq X \times X$ on the set of states of X as follows. For any $x_i, x_j \in X$,

$$(x_i, x_j) \in \Pi \quad \Leftrightarrow \quad (\exists q \in Q) x_i \in q \wedge x_j \in q.$$

Relation Π has the following properties.

(1) For all $x_i \in X$, $(x_i, x_i) \in \Pi$.
(2) For all $x_i, x_j \in X$, $(x_i, x_j) \in \Pi \Rightarrow (x_j, x_i) \in \Pi$.
(3) In general, $(x_i, x_j) \in \Pi \wedge (x_j, x_k) \in \Pi \not\Rightarrow (x_i, x_k) \in \Pi$.
(4) $\theta(s_i) = \theta(s_j) \Rightarrow (f(x_0, s_i), f(x_0, s_j)) \in \Pi$.

To calculate relation Π from its definition, we need to construct G_{obs}, which in the worst-case is of exponential complexity in terms of $|X|$, the number of states in X. One of the contributions of this paper is to present a new algorithm of polynomial complexity in $|X|$ for the calculation of Π.

2.1. The notion of state clusters

Definition. For $x \in X$, the "*cluster*" of x is defined as

$$c(x) := \begin{cases} UR(\{x\}) & \text{if } ((\exists x' \in X)(\exists e \in E) f(x', e) \\ & \qquad = x \wedge \mathscr{I}(x', e) = 1) \vee x = x_0; \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Thus, $c(x)$ is defined when state x is entered by an observable transition or it is the initial state. When it is defined, $c(x)$ is the "unobservable reach" from state x.

Denote the set of all possible clusters as

$$\Phi := \{c(x) : x \in X \wedge c(x) \text{is defined}\}.$$

Clearly, the number of clusters is less than $|X|$. We enumerate the set of clusters as

$$\Phi := \{\phi_0, \phi_1, \phi_2, \ldots, \phi_N\},$$

where we set $\phi_0 = UR(x_0)$.

Next, we define a function $\mathscr{R} : 2^X \times E \to 2^X$ as follows. For $X' \subseteq X,\ e \in E$,

$$\mathscr{R}(X', e) := \{x \in X : (\exists\ x' \in X') f(x', e) = x \wedge \mathscr{I}(x', e) = 1\}.$$