PROJECT REPORT

SEAN SHAHKARAMI

1. Introduction

In general, given a network of systems modeled as a directed graph, graphs fibration provides a way to understand when nodes on the network undergo synchronized behavior. The goal of this paper is to make this intuition more formal in the case of a network of hybrid systems by adapting a result from the paper *Networks of Hybrid Systems* from the language of category theory to hybrid automata. The structure of the paper will be to introduce graph fibrations and a few of their properties, define a hybrid networks and their semantics and finally state and show the final result.

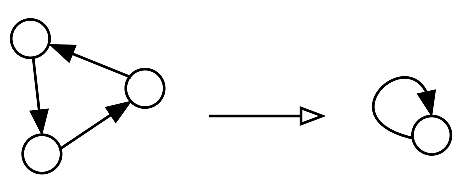
2. Graph Fibrations

In this section, we'll introduce graph fibrations, the main tool used in this paper. The intuition which we would like to formalize is that given a graph fibration from a source network onto a target network with a strictly smaller number of nodes, we'll be able to choose starting states so that nodes in the source network undergo synchronized behavior. Concisely, *proper graph fibrations yield synchronous behavior*.

Definition 1. A graph fibration is a graph homomorphism $\phi : \tilde{G} \to G$ which satisfies the property that for any vertex $\tilde{v} \in \tilde{G}$ and for any edge $e \in G$ with $t(e) = \phi(v)$ there is a *unique* edge $\tilde{e} \in \tilde{G}$ such that $\phi(\tilde{e}) = e$ and $t(\tilde{e}) = \tilde{v}$.

In other words, we're able to uniquely lift edges from the target graph to the source graph via a graph fibration.

Example 2. The graph homomorphism defined by sending the following three vertices to a single vertex is a graph fibration. In this case, it's easy to check the unique edge lifting property.



One of the key properties of graph fibrations is that they preserve the local incoming edge structure between graphs. Although this will be used in the main result, it is generally useful criteria for whether a graph may admit a graph fibration or not.

Lemma 3. *If $\phi : \tilde{G} \to G$ is a fibration then for each $\tilde{v} \in \tilde{G}$ then ϕ provides a bijection between incoming edges of $\tilde{v}$ and incoming edges of $\phi(\tilde{v})$.*

Proof. For any vertex $\tilde{v} \in \tilde{G}$ and any edge $\tilde{e} \in G$ with $t(\tilde{e}) = \tilde{v}$, we have $t(\phi(\tilde{e})) = \phi(v)$. Since $\tilde{e}$ is uniquely determined by the lifting property ϕ is an injective map from edges ending at $\tilde{v}$ to edges ending at $\phi(\tilde{v})$. Since we can also lift any edge e ending at $\phi(\tilde{v})$ to one ending at $\tilde{v}$, ϕ is also surjective. □

To make this more concise, we'll introduce the notation $I(v)$ to mean the set of incoming neighbors of v. In this notation, the lemma just says $I(\tilde{v})$ and $I(\phi(\tilde{v}))$ are in one-to-one correspondence.

Finally, before moving onto hybrid networks, one last interesting property of graph fibrations which will further "justify" (I mean in the sense that, our main concern is already finding proper graph fibrations so this is only a tangential remark.) us to formulate the result only for surjective graph fibrations is the following

Lemma 4. *If $\phi : \tilde{G} \to G$ is a fibration with $\tilde{G}$ nonempty and G strongly-connected then ϕ is surjective.*

Proof. Since $\tilde{G}$ is nonempty, there exists $\tilde{v} \in \tilde{G}$. Let $v = \phi(\tilde{v})$. For any $w \in G$, there exists a path from v to w and a path from w to v. Each of these paths lifts edge-wise to paths in $\tilde{G}$ projecting to the paths in G. The vertices along the lifted paths cover the vertices in original paths. In particular, w is covered. □

Intuitively, this this tells us a graph fibration mapping onto a strongly-connected component covers the whole component and because the strongly-connected components form an acyclic graph there can be no "feedback" behavior between systems is different components of the network.

3. Hybrid Networks

In this section we introduce hybrid networks. The semantics of a hybrid network will be defined by indirectly by introducing an associated, ordinary hybrid system.

Definition 5. A *hybrid network* consists of

- A directed graph G.
- An assignment $\mathcal{A}_v = (X_v, Q_v, \Theta_v, A_v, D_v, T_v)$ of hybrid automata to each vertex $v \in G$ with the modification that the continuous dynamics of $\mathcal{A}_v$ is allowed to depend on the state Q_w of $\mathcal{A}_w$ for each incoming neighbor w of v.

To a hybrid network $(G, \mathcal{A}_v)$ we associate an ordinary hybrid automata $G_{||}$ defined by

- The variables are $X_{||} := \coprod_{v \in G} X_v$.
- The state space is $Q_{||} := \prod_{v \in G} Q_v$.
- The start states are $\Theta_{||} := \prod_{v \in G} \Theta_v$.
- The actions are $A_{||} := \bigcup_{v \in G} A_v$.
- For $x, y \in Q_{||}, a \in A_{||}$ we have $x \xrightarrow{a} y$ iff for each $v \in G$ either (1) $a \in A_v$ and $x \lceil X_v \xrightarrow{a} y \lceil X_v$ or (2) $a \notin A_v$ and $x \lceil X_v = y \lceil X_v$.

- Trajectories are solutions to the differential equations (now without parameters).

4. Main Result

Now that we have defined graph fibrations and hybrid networks, we introduce and prove the main result.

Theorem 6. *Given hybrid networks $(\tilde{G}, \tilde{\mathcal{A}}_v)$ and $(G, \mathcal{A}_v)$ and a surjective graph fibration $\phi : \tilde{G} \to G$ such that $\tilde{\mathcal{A}v} = \mathcal{A}_{\phi(\tilde{v})}$, there exists a forward simulation of $G_{||}$ into $\tilde{G}_{||}$ such that under any execution of $G_{||}$ ending at state q, there is a related execution in $\tilde{G}_{||}$ ending at $\tilde{q}$ such that for each $\tilde{v} \in \tilde{G}$ with $\tilde{q}_{\tilde{v}} = q_{\phi(\tilde{v})}$.*

Proof. Let R be the relation such that the state q in $G_{||}$ is related to the state $\tilde{q}$ in $\tilde{G}_{||}$ where $\tilde{q}_{\tilde{v}} = q_{\phi(\tilde{v})}$.

It's clear that start states in $G_{||}$ are related to the start states in $\tilde{G}_{||}$.

We now check that transitions in $G_{||}$ are related to transitions in $\tilde{G}_{||}$. Let $q \xrightarrow{a} p$ be a transition in $G_{||}$ and let $\tilde{q}, \tilde{p}$ be the states in $\tilde{G}_{||}$ related to q, p respectively. For each $v \in G$ and each $\tilde{v} \in \phi^{-1}(v)$ we have either (1) if $a \in A_{\tilde{v}}$ then since $q_v \xrightarrow{a} p_v$ and $\tilde{\mathcal{A}v} = \mathcal{A}_{\phi(\tilde{v})}$ we have $\tilde{q}_{\tilde{v}} \xrightarrow{a} \tilde{p}_{\tilde{v}}$. (2) if $a \notin A_{\tilde{v}}$ then since $q_v = p_v$ and $\tilde{\mathcal{A}v} = \mathcal{A}_{\phi(\tilde{v})}$ we have $\tilde{q}_{\tilde{v}} = \tilde{p}_{\tilde{v}}$. Thus, $\tilde{q} \xrightarrow{a} \tilde{p}$ in $\tilde{G}_{||}$.

We now check that trajectories in $G_{||}$ are related to trajectories in $\tilde{G}_{||}$. To see this, let q be a state in $G_{||}$ and $\tilde{q}$ be the related state in $\tilde{G}_{||}$. From lemma 3, we have that for any $v \in G$ and any $\tilde{v} \in \phi^{-1}(v)$, the dependence of the continuous dynamics of $\tilde{\mathcal{A}}_{\tilde{v}}$ on the neighboring states $\tilde{\mathcal{A}}_{\tilde{w}}$ for each $\tilde{w} \in I(\tilde{v})$ is the same as the dependence of $\mathcal{A}_v$ on $\mathcal{A}_w$ for each $w \in I(v)$. Since $\tilde{\mathcal{A}v} = \mathcal{A}_{\phi(\tilde{v})}$ their ODEs are same and since $\tilde{q}_{\tilde{v}} = q_v$ and for each $\tilde{w} \in I(\tilde{v})$, $\tilde{q}_{\tilde{w}} = q_{\phi(\tilde{w})}$ the initial conditions are the same. This ensures $\mathcal{A}_v$ and $\tilde{\mathcal{A}}_{\tilde{v}}$ evolve identically which implies the final state of the trajectory in $G_{||}$ is related to the final state of the trajectory in $\tilde{G}_{||}$. □

This formalizes the intuition that proper graph fibrations yield synchronous behavior. As long as we can find nodes in the source network above a single base node in the target network, we're able to lift an execution in a way where the nodes will stay in the same state the entire time.

5. Closing Thoughts

Although this is a simple result, a couple questions follow naturally from this. First, is there an efficient way to find graph fibrations in light of lemma 3 and 4? Second, the correspondence between synchronized behavior and graph fibrations is not complete. That is, if there is some kind of synchronized behavior can we encode it as a graph fibration? I haven't yet given too much thought to either of these questions. While both answers may turn out to have a quick no, they may be interesting considerations.

References

[NOHS] Eugene Lerman, Sayan Mitra. Networks of Hybrid Systems.
[TIOA] Dilsun Kaynar, Nancy Lynch, Roberto Segala, Frits Vaandrager. The Theory of Timed I/O Automata.

[MFC] Glynn Winskel, Nielsen Mogens. Models for Concurrency.
[FOG] Paolo Boldi, Sebastiano Vigna. Fibrations of Graphs.