Decidable Topologies for Communicating Automata

Sriram Balasubramanian   Adwait Godbole

October 2019

Trajectory

Labelled Transition System

Definition

A *labelled transition system* is a tuple $\mathcal{A} = \langle S, S_I, S_F, A, \rightarrow \rangle$ where

Labelled Transition System

Definition

A *labelled transition system* is a tuple $\mathcal{A} = \langle S, S_I, S_F, A, \rightarrow \rangle$ where

1. S is a set of states
2. S_I and S_F are initial and final state sets

Labelled Transition System

Definition

A *labelled transition system* is a tuple $\mathcal{A} = \langle S, S_I, S_F, A, \rightarrow \rangle$ where

1. S is a set of states
2. S_I and S_F are initial and final state sets
3. A is the finite set of actions

Labelled Transition System

Definition

A *labelled transition system* is a tuple $\mathcal{A} = \langle S, S_I, S_F, A, \rightarrow \rangle$ where

1. S is a set of states
2. S_I and S_F are initial and final state sets
3. A is the finite set of actions
4. $\rightarrow \ \subseteq \ S \times A \times S$ is a labelled transition relation

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes
2. C is a finite set of channels

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes
2. C is a finite set of channels
3. M is the (finite) message alphabet

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes
2. C is a finite set of channels
3. M is the (finite) message alphabet
4. src and dst are functions from $C \to P$ marking the source and the destination for each channel

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes
2. C is a finite set of channels
3. M is the (finite) message alphabet
4. src and dst are functions from $C \rightarrow P$ marking the source and the destination for each channel
5. msg is the function that assigns to each C its message alphabet (a subset of M)

Communication Topology

Definition

A *communication topology* is a tuple $\mathcal{T} = \langle P, C, M, \mathrm{src}, \mathrm{dst}, \mathrm{msg} \rangle$ where

1. P is a finite set of communicating processes
2. C is a finite set of channels
3. M is the (finite) message alphabet
4. src and dst are functions from $C \to P$ marking the source and the destination for each channel
5. msg is the function that assigns to each C its message alphabet (a subset of M)

System of Communicating Processes

A system of communicating processes is a pair $\mathcal{S} = \langle \mathcal{T}, \{\mathcal{A}^p\}_{p \in P} \rangle$ of a topology and its constituent processes

Bag v.s. FIFO v.s Lossy Channels

A channel can be of three types:

1. Bag channel - where the messages do not have any order
2. FIFO channel - where the channel is a FIFO queue with total order on the messages per channel
3. Lossy channel - where messages can be arbitrarily dropped from the channel

A few preliminary results

Decidability of Bag-*only* systems

A communicating system where all channels are Bag type channels is decidable

A few preliminary results

Decidability of Bag-*only* systems

A communicating system where all channels are Bag type channels is decidable

Proof idea

We can convert this problem into a Petri Net reachability problem. Note that with the order on messages removed, each channel can be viewed as a multiset of messages.

A few preliminary results

Decidability of Bag-*only* systems

A communicating system where all channels are Bag type channels is decidable

Proof idea

We can convert this problem into a Petri Net reachability problem. Note that with the order on messages removed, each channel can be viewed as a multiset of messages.

1. *For each channel-message pair* (c, m)*, construct a place* s_m^c *in the Petri-Net storing tokens corresponding to the number of such* m *type messages in* c*.*

A few preliminary results

Decidability of Bag-*only* systems

A communicating system where all channels are Bag type channels is decidable

Proof idea

We can convert this problem into a Petri Net reachability problem. Note that with the order on messages removed, each channel can be viewed as a multiset of messages.

1. *For each channel-message pair (c, m), construct a place s_m^c in the Petri-Net storing tokens corresponding to the number of such m type messages in c.*
2. *The control state can be modelled with additional states.*

A few preliminary results

Decidability of Bag-*only* systems

A communicating system where all channels are Bag type channels is decidable

Proof idea

We can convert this problem into a Petri Net reachability problem. Note that with the order on messages removed, each channel can be viewed as a multiset of messages.

1. *For each channel-message pair (c, m), construct a place s_m^c in the Petri-Net storing tokens corresponding to the number of such m type messages in c.*
2. *The control state can be modelled with additional states.*
3. *For action a corresponding to $c?m$ $(c!m)$ is translated into a transition t with s_m^c being one of the input (output) places; the others being those for the control state.*

A few preliminary results

Decidability of *unary*-channel systems

A communicating system where all channels are unary (having $|\text{msg}(c)| = 1$) is decidable

A few preliminary results

Decidability of *unary*-channel systems

A communicating system where all channels are unary (having $|\text{msg}(c)| = 1$) is decidable

Unary channels are special cases of Bag channels

A few preliminary results

Decidability of perfect FIFO channel systems

A communicating system where all channels are perfect FIFOs is decidable if and only if the topology is a polyforest, i.e, a directed acyclic graph whose underlying undirected graph is a tree.

A few preliminary results

Decidability of perfect FIFO channel systems

A communicating system where all channels are perfect FIFOs is decidable if and only if the topology is a polyforest, i.e, a directed acyclic graph whose underlying undirected graph is a tree.

Proof idea

If *is proved later.* **Only if** *is as follows:*

A few preliminary results

Decidability of perfect FIFO channel systems

A communicating system where all channels are perfect FIFOs is decidable if and only if the topology is a polyforest, i.e, a directed acyclic graph whose underlying undirected graph is a tree.

Proof idea

If *is proved later.* **Only if** *is as follows:*

1. *We prove that reachability in all (two) topologies consisting of two processes which have a cycle in the underlying undirected graph is undecidable*

A few preliminary results

Decidability of perfect FIFO channel systems

A communicating system where all channels are perfect FIFOs is decidable if and only if the topology is a polyforest, i.e, a directed acyclic graph whose underlying undirected graph is a tree.

Proof idea

If *is proved later.* **Only if** *is as follows:*

1. *We prove that reachability in all (two) topologies consisting of two processes which have a cycle in the underlying undirected graph is undecidable*
2. *Proceed by induction to prove undecidability for topologies with an n length cycle if the cycle has two consecutive channels in the same direction*

A few preliminary results

Decidability of perfect FIFO channel systems

A communicating system where all channels are perfect FIFOs is decidable if and only if the topology is a polyforest, i.e, a directed acyclic graph whose underlying undirected graph is a tree.

Proof idea

If *is proved later.* **Only if** *is as follows:*

1. *We prove that reachability in all (two) topologies consisting of two processes which have a cycle in the underlying undirected graph is undecidable*
2. *Proceed by induction to prove undecidability for topologies with an n length cycle if the cycle has two consecutive channels in the same direction*
3. *If no two consecutive channels in the n length cycle are in the same direction, prove by simulating PCP.*

Trajectory

Trajectory

Systems with FIFO + Lossy channels

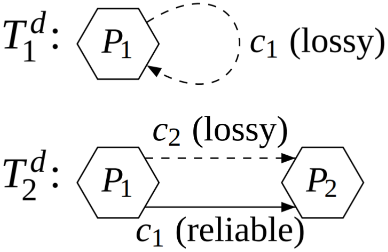


Figura: Two simple systems

Consider the systems above. We have seen that state reachability for T_1^d is decidable. What about T_2^d?

Post's Embedding Problem

We consider the following variant(s) of Post's Correspondence Problem.

Post's Embedding Problem (PEP)

For two finite alphabets Σ and Γ, and two morphisms $u, v : \Sigma^* \to \Gamma^*$, does there exist a $\sigma \in \Sigma^+$ such that $u(\sigma) \sqsubseteq v(\sigma)$?

Post's Embedding Problem

We consider the following variant(s) of Post's Correspondence Problem.

Post's Embedding Problem (PEP)

For two finite alphabets Σ and Γ, and two morphisms $u, v : \Sigma^* \to \Gamma^*$, does there exist a $\sigma \in \Sigma^+$ such that $u(\sigma) \sqsubseteq v(\sigma)$?

Post's Embedding Problem (PEP^{reg})

For two finite alphabets Σ and Γ, and two morphisms $u, v : \Sigma^* \to \Gamma^*$, and a regular language $R \subseteq \Sigma^*$, does there exist a $\sigma \in R$ such that $u(\sigma) \sqsubseteq v(\sigma)$?

Post's Embedding Problem

We consider the following variant(s) of Post's Correspondence Problem.

Post's Embedding Problem (PEP)

For two finite alphabets Σ and Γ, and two morphisms $u, v : \Sigma^* \to \Gamma^*$, does there exist a $\sigma \in \Sigma^+$ such that $u(\sigma) \sqsubseteq v(\sigma)$?

Post's Embedding Problem (PEP^{reg})

For two finite alphabets Σ and Γ, and two morphisms $u, v : \Sigma^* \to \Gamma^*$, and a regular language $R \subseteq \Sigma^*$, does there exist a $\sigma \in R$ such that $u(\sigma) \sqsubseteq v(\sigma)$?

Both the above problems are decidable. Decidability of PEP is trivial. PEP^{reg} on the other hand is non-trivial.

Decidability of T_2^d

Let the FIFO and lossy channels be denoted by f and l respectively. Also let $\alpha(\delta)$ and $\beta(\delta)$ denote the messages read or written in transition δ. Let Δ_1 and Δ_2 denote the transitions of the sender and the receiver respectively.

Decidability of T_2^d

Let the FIFO and lossy channels be denoted by f and l respectively. Also let $\alpha(\delta)$ and $\beta(\delta)$ denote the messages read or written in transition δ. Let Δ_1 and Δ_2 denote the transitions of the sender and the receiver respectively.

Proof idea

1. *Encode as a PEP^{reg} instance*

Decidability of T_2^d

Let the FIFO and lossy channels be denoted by f and l respectively. Also let $\alpha(\delta)$ and $\beta(\delta)$ denote the messages read or written in transition δ. Let Δ_1 and Δ_2 denote the transitions of the sender and the receiver respectively.

Proof idea

1. *Encode as a PEP^{reg} instance*
2. *Unidirectionality allows for reordering of actions such that first transitions in Δ_1 are performed followed by those in Δ_2*

Decidability of T_2^d

Let the FIFO and lossy channels be denoted by f and l respectively. Also let $\alpha(\delta)$ and $\beta(\delta)$ denote the messages read or written in transition δ. Let Δ_1 and Δ_2 denote the transitions of the sender and the receiver respectively.

Proof idea

1. *Encode as a PEP^{reg} instance*
2. *Unidirectionality allows for reordering of actions such that first transitions in Δ_1 are performed followed by those in Δ_2*
3. *Define u and v as follows:*
   - $u(\delta) = \beta(\delta)$ *for* $\delta \in \Delta_2$ *and* ϵ *otherwise*
   - $v(\delta) = \alpha(\delta)$ *for* $\delta \in \Delta_1$ *and* ϵ *otherwise*

Decidability of T_2^d

Let the FIFO and lossy channels be denoted by f and l respectively. Also let $\alpha(\delta)$ and $\beta(\delta)$ denote the messages read or written in transition δ. Let Δ_1 and Δ_2 denote the transitions of the sender and the receiver respectively.

Proof idea

1. *Encode as a PEP^{reg} instance*
2. *Unidirectionality allows for reordering of actions such that first transitions in Δ_1 are performed followed by those in Δ_2*
3. *Define u and v as follows:*
   - $u(\delta) = \beta(\delta)$ *for* $\delta \in \Delta_2$ *and* ϵ *otherwise*
   - $v(\delta) = \alpha(\delta)$ *for* $\delta \in \Delta_1$ *and* ϵ *otherwise*
4. *Let* $R = (R_1 \bowtie R_2) \cap R_3^*$ *where*
   - R_1 *and* R_2 *capture accepting state reachability of sender and receiver, i.e.* $s_0 \xrightarrow{R_{1/2}} s_f$
   - R_3 *captures language where writes on channel* f *are matched by a read, i.e.* $R_3 = \{\delta\gamma | \delta\gamma \in \Delta_1\Delta_2 \text{ with } \alpha(\delta) = \alpha(\gamma)\}$

Systems with FIFO + Lossy channels

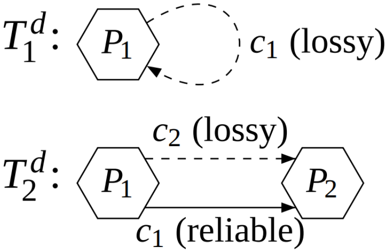


Figura: Two simple systems

Thus both are decidable. What if the channels in T_2^d were in opposite directions? We will come back to this case later.

Some Undecidable Cases

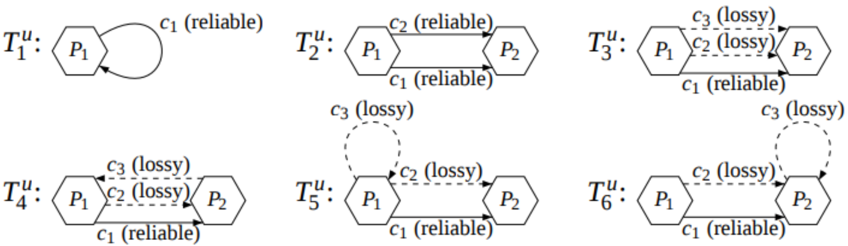


Figura: Undecidable Topologies

Proof for all of these is by PCP with some nice tricks!

Trajectory

Towards a Complete Characterisation

1. $T - c$ is the topology obtained by deleting channel c

Towards a Complete Characterisation

1. $T - c$ is the topology obtained by deleting channel c
2. $T[P_1{=}P_2]$ is the topology obtained by combining processes P_1 and P_2

Towards a Complete Characterisation

1. $T - c$ is the topology obtained by deleting channel c
2. $T[P_1{=}P_2]$ is the topology obtained by combining processes P_1 and P_2
3. T/c is the topology obtained by contracting channel c

Towards a Complete Characterisation

1. $T - c$ is the topology obtained by deleting channel c
2. $T[P_1{=}P_2]$ is the topology obtained by combining processes P_1 and P_2
3. T/c is the topology obtained by contracting channel c

Definition 1

A channel c is *essential* if all directed paths from $src(c)$ to $dst(c)$ contain c. In particular, $src(c) \neq dst(c)$.

Towards a Complete Characterisation

1. $T - c$ is the topology obtained by deleting channel c
2. $T[P_1{=}P_2]$ is the topology obtained by combining processes P_1 and P_2
3. T/c is the topology obtained by contracting channel c

Definition 1

A channel c is *essential* if all directed paths from $src(c)$ to $dst(c)$ contain c. In particular, $src(c) \neq dst(c)$.

Theorem 1

If c is an essential channel, then every run that starts and ends with c empty is causal-equivalent to a run that is synchronous for c.

Proof of Theorem 1

Proof by induction on length of runs:

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case
- If length of run is n, and first action does not write on c, we are done.

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case
- If length of run is n, and first action does not write on c, we are done.
- Else, the run is of the form: $\rho = x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1 \cdot x_2 \xrightarrow{\text{c?m}} x_2' \cdot \rho_2$

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case
- If length of run is n, and first action does not write on c, we are done.
- Else, the run is of the form: $\rho = x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1 \cdot x_2 \xrightarrow{\text{c?m}} x_2' \cdot \rho_2$
- Let $p = src(c)$, $q = dst(c)$. Then, we can reorder $x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1$ to $z_1 \cdot x_1 \xrightarrow{\text{c!m}} x_1' \cdot z_2$ where z_1 does not contain any action in p and z_2 does not contain any action in q, because the channel c is essential.

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case
- If length of run is n, and first action does not write on c, we are done.
- Else, the run is of the form: $\rho = x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1 \cdot x_2 \xrightarrow{\text{c?m}} x_2' \cdot \rho_2$
- Let $p = src(c)$, $q = dst(c)$. Then, we can reorder $x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1$ to $z_1 \cdot x_1 \xrightarrow{\text{c!m}} x_1' \cdot z_2$ where z_1 does not contain any action in p and z_2 does not contain any action in q, because the channel c is essential.
- Therefore, $z_2 \cdot x_2 \xrightarrow{\text{c?m}} x_2'$ can be reordered into $x_2 \xrightarrow{\text{c?m}} x_2' \cdot z_2$. Now, send and receive of m is synchronized. Applying induction hypothesis on ρ_2, we get the result.

Proof of Theorem 1

Proof by induction on length of runs:

- Base case: Length of run is 0, trivial case
- If length of run is n, and first action does not write on c, we are done.
- Else, the run is of the form: $\rho = x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1 \cdot x_2 \xrightarrow{\text{c?m}} x_2' \cdot \rho_2$
- Let $p = src(c)$, $q = dst(c)$. Then, we can reorder $x_1 \xrightarrow{\text{c!m}} x_1' \cdot \rho_1$ to $z_1 \cdot x_1 \xrightarrow{\text{c!m}} x_1' \cdot z_2$ where z_1 does not contain any action in p and z_2 does not contain any action in q, because the channel c is essential.
- Therefore, $z_2 \cdot x_2 \xrightarrow{\text{c?m}} x_2'$ can be reordered into $x_2 \xrightarrow{\text{c?m}} x_2' \cdot z_2$. Now, send and receive of m is synchronized. Applying induction hypothesis on ρ_2, we get the result.

Using Theorem 1, we can prove that effective channels can be replaced using two bag channels without affecting reachability and other important properties.

Decidability of polyforest topologies

In a polyforest, all channels are essential, or else there would be a cycle in their undirected graphs. Therefore, all channels can be synchronized and replaced by two bags. Reachability of all bag systems is decidable, and therefore, reachability in polyforest systems is decidable.

Decidability by Fusion

Decidability by Fusion

Let c be a reliable channel in T. Then

1. T has decidable reachability if T/c has
2. If in addition c is a reachable channel then T/c has decidable reachability if T has

Decidability by Fusion

Decidability by Fusion

Let c be a reliable channel in T. Then

1. T has decidable reachability if T/c has
2. If in addition c is a reachable channel then T/c has decidable reachability if T has

Proof sketch (Part 1)

This is a simple consequence of Theorem 1.

Decidability by Fusion

Decidability by Fusion

Let c be a reliable channel in T. Then

1. T has decidable reachability if T/c has
2. If in addition c is a reachable channel then T/c has decidable reachability if T has

Proof sketch (Part 1)

This is a simple consequence of Theorem 1.

1. Consider S as the system with topology T/c

Decidability by Fusion

Decidability by Fusion

Let c be a reliable channel in T. Then

1. T has decidable reachability if T/c has
2. If in addition c is a reachable channel then T/c has decidable reachability if T has

Proof sketch (Part 1)

This is a simple consequence of Theorem 1.

1. Consider S as the system with topology T/c
2. Due to Theorem 1, S only needs to simulate runs where c is synchronous

Decidability by Fusion

Decidability by Fusion

Let c be a reliable channel in T. Then

1. T has decidable reachability if T/c has
2. If in addition c is a reachable channel then T/c has decidable reachability if T has

Proof sketch (Part 1)

This is a simple consequence of Theorem 1.

1. Consider S as the system with topology T/c
2. Due to Theorem 1, S only needs to simulate runs where c is synchronous
3. Hence the channel states can be assimilated into the state corresponding to P_1 an P_2 and the result follows due to decidability of S

Some Examples of Fusion

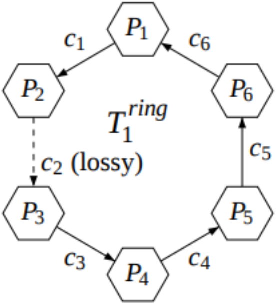


Figura: T_1^{ring}

What can we say about control state reachability of T_1^{ring}?

Some Examples of Fusion

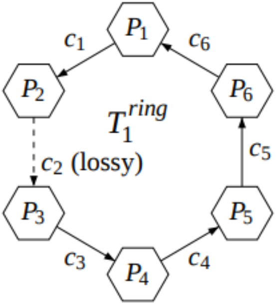


Figura: T_1^{ring}

What can we say about control state reachability of T_1^{ring}?

T_1^{ring} is decidable by fusing all the reliable channels!

Some Examples of Fusion

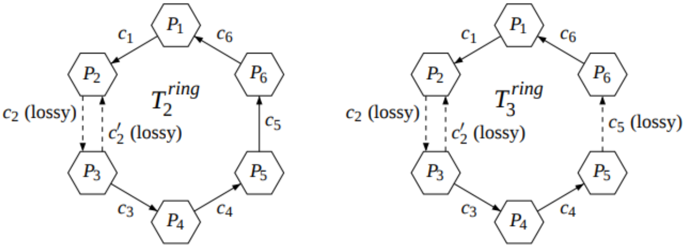


Figura: T_2^{ring} and T_3^{ring}

What about T_2^{ring} and T_3^{ring}?

Some Examples of Fusion

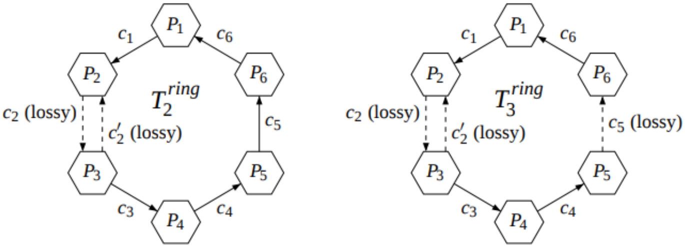


Figura: T_2^{ring} and T_3^{ring}

What about T_2^{ring} and T_3^{ring}?

T_2^{ring} is undecidable. T_3^{ring} is decidable.

Decidability by Lossy Splitting

Lossy Splitting

A topology T that can be split into T_1 and T_2 by deleting unidirectional lossy channels between T_1 and T_2 is denoted as $T_1 \triangleright T_2$

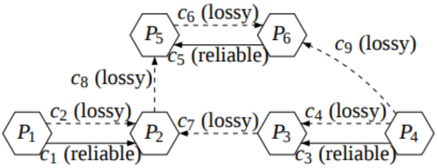


Figura: A topology that can be split into 3 components $(T_2 \triangleright (T_1 \triangleright T_3))$

Decidability by Lossy Splitting

Lossy Splitting

A topology T that can be split into T_1 and T_2 by deleting unidirectional lossy channels between T_1 and T_2 is denoted as $T_1 \triangleright T_2$

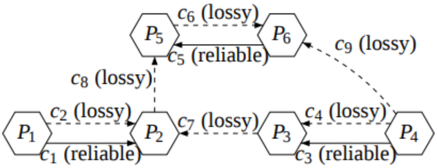


Figura: A topology that can be split into 3 components $(T_2 \triangleright (T_1 \triangleright T_3))$

Decidability by Splitting

Reachability is decidable for $T_1 \triangleright T_2$ iff it is decidable for both T_1 and T_2

Complete Characterisation of FIFO + Lossy Systems

Complete Characterisation Theorem

A network topology with FIFO + Lossy channels is decidable iff it can be reduced to T_2^d or LCS using fusion and splitting only

Complete Characterisation of FIFO + Lossy Systems

Complete Characterisation Theorem

A network topology with FIFO + Lossy channels is decidable iff it can be reduced to T_2^d or LCS using fusion and splitting only

Proof idea

The 'if' direction follows from the decidability by fusion and splitting that we saw earlier. The 'only if' direct requires an involved case analysis on the irreducible topology:

Complete Characterisation of FIFO + Lossy Systems

Complete Characterisation Theorem

A network topology with FIFO + Lossy channels is decidable iff it can be reduced to T_2^d or LCS using fusion and splitting only

Proof idea

The 'if' direction follows from the decidability by fusion and splitting that we saw earlier. The 'only if' direct requires an involved case analysis on the irreducible topology:

1. *There exists a reliable channel c_r linking some nodes A and B*

Complete Characterisation of FIFO + Lossy Systems

Complete Characterisation Theorem

A network topology with FIFO + Lossy channels is decidable iff it can be reduced to T_2^d or LCS using fusion and splitting only

Proof idea

The 'if' direction follows from the decidability by fusion and splitting that we saw earlier. The 'only if' direct requires an involved case analysis on the irreducible topology:

1. *There exists a reliable channel c_r linking some nodes A and B*
2. *Must also exist an alternative path θ with atleast one lossy channel*

Complete Characterisation of FIFO + Lossy Systems

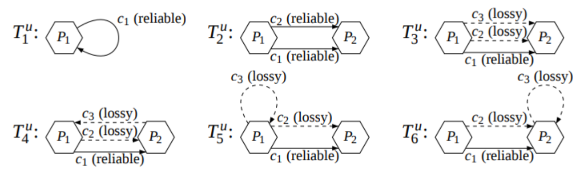
Complete Characterisation Theorem

A network topology with FIFO + Lossy channels is decidable iff it can be reduced to T_2^d or LCS using fusion and splitting only

Proof idea

The 'if' direction follows from the decidability by fusion and splitting that we saw earlier. The 'only if' direct requires an involved case analysis on the irreducible topology:

1. *There exists a reliable channel c_r linking some nodes A and B*
2. *Must also exist an alternative path θ with atleast one lossy channel*



Trajectory

Trajectory

1. Preliminaries
2. FIFO + Lossy Channel Systems
   - Some Basic Systems
   - Complete Characterisation
3. FIFO + Bag Channel Systems
   - Some Techniques
   - Examples
   - Decidability Characterization

Systems with Bags + FIFO channels

When both types of channels are present in the topology, we primarily use two reduction techniques to show that reachability is decidable in the concerned topology.

Systems with Bags + FIFO channels

When both types of channels are present in the topology, we primarily use two reduction techniques to show that reachability is decidable in the concerned topology.

Synchronization

Replace a FIFO from p to q by two channels, one from p to q and the other from q to p. The communication on these two channels is *synchronized*, as the sender waits for an acknowledgement(ACK) from the receive before sending another message.

Systems with Bags + FIFO channels

When both types of channels are present in the topology, we primarily use two reduction techniques to show that reachability is decidable in the concerned topology.

Synchronization

Replace a FIFO from p to q by two channels, one from p to q and the other from q to p. The communication on these two channels is *synchronized*, as the sender waits for an acknowledgement(ACK) from the receive before sending another message.

Splitting

Redirect a bag channel from p to q to a new process r, with a new bag channel from q to r. The reads of q are now writes on the new channel. The new process just matches messages of both channels.

Synchronization

Recap: Essential channels and Synchronization Theorem

Definition 1

A channel c is *essential* if all directed paths from $src(c)$ to $dst(c)$ contain c. In particular, $src(c) \neq dst(c)$.

Theorem 1

If c is an essential channel, then every run that starts and ends with c empty is causal-equivalent to a run that is synchronous for c.

Splitting Revisited

Definition 2

A channel c is *reversible* if there is a directed path from its destination $dst(c)$ to its source $src(c)$. A channel is *irreversible* if it is not reversible.

Definition 3

A run $(x_0, a_1, x_1, \ldots a_n, x_n)$ is *causal* for a given process p if $q \stackrel{*}{\Rightarrow} p$ for every process q such that $a_i \in A_q$ and $a_j \in A_p$ for some $1 \leq i < j \leq n$.

Splitting Revisited

Definition 2

A channel c is *reversible* if there is a directed path from its destination $dst(c)$ to its source $src(c)$. A channel is *irreversible* if it is not reversible.

Definition 3

A run $(x_0, a_1, x_1, \ldots a_n, x_n)$ is *causal* for a given process p if $q \overset{*}{\Rightarrow} p$ for every process q such that $a_i \in A_q$ and $a_j \in A_p$ for some $1 \leq i < j \leq n$.

Lemma 1

Given a process p, every run is causal-equivalent to a run that is causal for p.

Splitting Revisited

Definition 2

A channel c is *reversible* if there is a directed path from its destination $dst(c)$ to its source $src(c)$. A channel is *irreversible* if it is not reversible.

Definition 3

A run $(x_0, a_1, x_1, \ldots a_n, x_n)$ is *causal* for a given process p if $q \overset{*}{\Rightarrow} p$ for every process q such that $a_i \in A_q$ and $a_j \in A_p$ for some $1 \leq i < j \leq n$.

Lemma 1

Given a process p, every run is causal-equivalent to a run that is causal for p.

Theorem 2

If c is an irreversible channel in $\mathcal{T}$, then it holds that $Reach(\mathcal{T}) \preceq Reach(\mathcal{U})$ where $\mathcal{U}$ results from the split of c in $\mathcal{T}$.

Proof of Theorem 2

We make a construction as before:

Proof of Theorem 2

We make a construction as before:

- Redirect a bag channel from p to q to a new process r, with a new bag channel from q to r. The reads of q are now writes on the new channel. Process r just matches messages of both channels.

Proof of Theorem 2

We make a construction as before:

- Redirect a bag channel from p to q to a new process r, with a new bag channel from q to r. The reads of q are now writes on the new channel. Process r just matches messages of both channels.
- Any accepting run in $\mathcal{T}$ is easily translated to an accepting run in $\mathcal{U}$.

Proof of Theorem 2

We make a construction as before:

- Redirect a bag channel from p to q to a new process r, with a new bag channel from q to r. The reads of q are now writes on the new channel. Process r just matches messages of both channels.
- Any accepting run in $\mathcal{T}$ is easily translated to an accepting run in $\mathcal{U}$.
- Consider any accepting run in $\mathcal{U}$. There exists a causal run for p causally equivalent to the given accepting run. In this run, all actions of p occur before any action in q.

Proof of Theorem 2

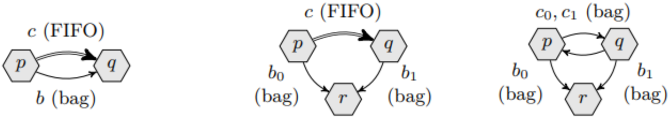
We make a construction as before:

- Redirect a bag channel from p to q to a new process r, with a new bag channel from q to r. The reads of q are now writes on the new channel. Process r just matches messages of both channels.
- Any accepting run in $\mathcal{T}$ is easily translated to an accepting run in $\mathcal{U}$.
- Consider any accepting run in $\mathcal{U}$. There exists a causal run for p causally equivalent to the given accepting run. In this run, all actions of p occur before any action in q.
- Thus, we can simulate this run exactly in $\mathcal{T}$, by replacing sends from p to r with corresponding sends from p to q, and replacing sends from q to r with reads by q from p.

Trajectory

1. Preliminaries
2. FIFO + Lossy Channel Systems
   - Some Basic Systems
   - Complete Characterisation
3. FIFO + Bag Channel Systems
   - Some Techniques
   - Examples
   - Decidability Characterization

An Example

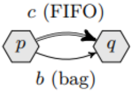


(a) Parallel channels b and c (b) Splitting b (c) Synchronising c

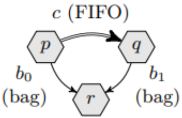
Figura: Some Two Channel Topologies

Now we'll discuss a topology to see the techniques seen earlier in action. Lets look at (a) in the figure above.

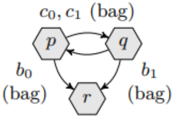
Some Examples



(a) Parallel channels b and c



(b) Splitting b



(c) Synchronising c

Figura: A Two Channel Topology

Some Examples

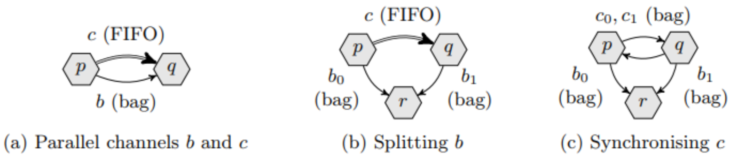


Figura: A Two Channel Topology

This example is decidable. Why?

Some Examples

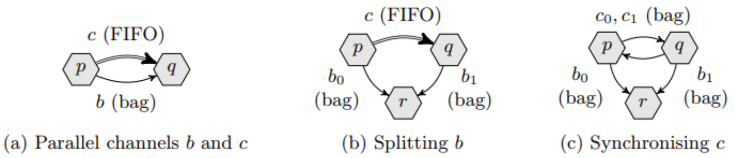
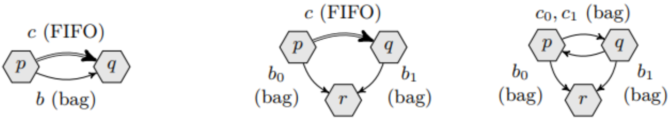


Figura: A Two Channel Topology

This example is decidable. Why?

- As discussed in the splitting idea, we can split the bag channel into two with an additional synchronization process r that matches the incoming tokens from b_0 and b_1.

Some Examples



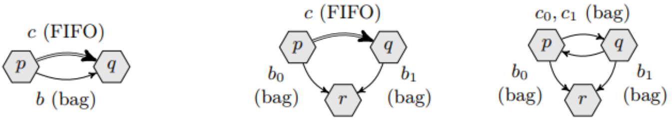
(a) Parallel channels b and c (b) Splitting b (c) Synchronising c

Figura: A Two Channel Topology

This example is decidable. Why?

- As discussed in the splitting idea, we can split the bag channel into two with an additional synchronization process r that matches the incoming tokens from b_0 and b_1.
- Now, the FIFO channel c is *essential* and hence communication can be synchronized on this channel.

Some Examples



(a) Parallel channels b and c (b) Splitting b (c) Synchronising c

Figura: A Two Channel Topology

This example is decidable. Why?

- As discussed in the splitting idea, we can split the bag channel into two with an additional synchronization process r that matches the incoming tokens from b_0 and b_1.
- Now, the FIFO channel c is *essential* and hence communication can be synchronized on this channel.

Thus we get a *bag*-only topology which is decidable.

Some Examples

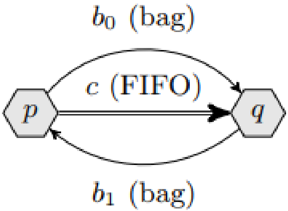


Figura: Another FIFO + Bag Topology

Some Examples

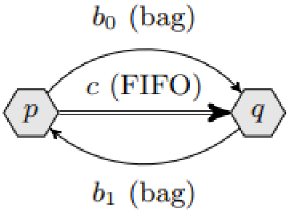


Figura: Another FIFO + Bag Topology

This example is undecidable. Even when b_0 and b_1 are unary! Why?

Some Examples

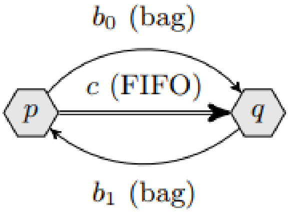


Figura: Another FIFO + Bag Topology

This example is undecidable. Even when b_0 and b_1 are unary! Why?

- This is possible by synchronizing the channels b_0 and b_1.

Some Examples

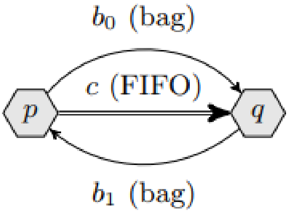


Figura: Another FIFO + Bag Topology

This example is undecidable. Even when b_0 and b_1 are unary! Why?

- This is possible by synchronizing the channels b_0 and b_1.
- This synchronized channel is used to decide which message can be read by q.

Trajectory

1. Preliminaries
2. FIFO + Lossy Channel Systems
   - Some Basic Systems
   - Complete Characterisation
3. FIFO + Bag Channel Systems
   - Some Techniques
   - Examples
   - Decidability Characterization

Definitions

Definition 1

Two processes p and q are *synchronizable* over D, written $p \approx_D q$, if there exists a directed path from p to q and a directed path from q to p, both using only channels in D.

Definitions

Definition 1

Two processes p and q are *synchronizable* over D, written $p \approx_D q$, if there exists a directed path from p to q and a directed path from q to p, both using only channels in D.

Definition 2

A *jumping circuit* is a sequence $(p_0, c_1, q_1, p_1, \ldots, c_n, q_n, p_n)$ of processes $p_i, q_i \in P$ and channels $c_i \in C$, with $n \geq 1$, such that $c_1, \ldots, c_n$ are pairwise distinct non-unary channels, $p_0 = p_n$, and $p_{i1} \xlongequal{c_i} q_i \approx_D p_i$ for all $1 \leq i \leq n$, where $D = C \setminus \{c_1, \ldots, c_n\}$. A *jumping cycle* is a jumping circuit such that $q_i \not\approx_D q_j$ for all $1 \leq i < j \leq n$.

Some observations

A few observations can be made:

- Every jumping circuit can be transformed into a jumping cycle.
- For every jumping cycle $p_0 \xlongequal{c_0} q_1 \approx_D p_1 \ldots \xlongequal{c_n} q_n \approx_D p_n$, there exist n pairwise disjoint subsets $D_1, \ldots, D_n$ of the set $D = C \setminus \{c_1, \ldots, c_n\}$ such that $q_i \approx_{D_i} p_i$ for all $1 \leq i \leq n$.

Characterization of decidability

Theorem

Given a topology $\mathcal{T}$, *Reach*($\mathcal{T}$) is decidable if, and only if, $\mathcal{T}$ has no jumping cycle.

Undecidability

Proof idea:

1. We fuse the set of supports (the D_is) for the directed paths between q_i and p_i one by one.
2. Just verify that after each step, the other D_is still act as supports for their respective channels, and at the end, we have a closed loop of FIFO channels.

Decidability

Lemma 1

Consider a topology $\mathcal{T}$ and an essential non-unary channel c therein. Let U be the topology obtained from $\mathcal{T}$ by adding an acknowledgement channel for $c(\overleftarrow{c})$. Then $\mathcal{T}$ contains a jumping cycle if U contains a jumping cycle.

Proof idea:

1. Assume $\mathcal{T}$ does not have jumping cycle but U does.
2. If $\overleftarrow{c}$ is a part of the directed path which synchronizes p_i and q_i, then since c is essential, c is also a part of the directed path.
3. This means that we can find a synchronization between q_i and $src(c)$ and $dst(c)$ and p_i.
4. This means here exists a jumping cycle in $\mathcal{T}$

Decidability

Definition 1

A topology $\mathcal{T}$ is *divided* if the destination of every irreversible unary channel is a sink (i.e., is not the source of some channel) and is not the destination of some non-unary channel.

Decidability

Definition 1

A topology $\mathcal{T}$ is *divided* if the destination of every irreversible unary channel is a sink (i.e., is not the source of some channel) and is not the destination of some non-unary channel.

Lemma 2

Consider a topology $\mathcal{T}$ and a non-unary channel c therein. If $\mathcal{T}$ is divided, then so is the topology resulting from the synchronisation of c in $\mathcal{T}$.

Lemma 3

If $\mathcal{T}$ is a divided topology with no jumping cycle, then every non-unary channel in $\mathcal{T}$ is essential.

Proof of Lemma 3

1. Assume that c is not essential. This means that there exists a directed path π from p to q that does not contain c.
2. We can show that c is irreversible, else there would be jumping cycle.
3. We can break down π into a sequence of reversible and irreversible channels, so

$$\pi = \chi_0 \cdot p_0 \Rightarrow q_1 \cdot \chi_1 \ldots p_{n1} \Rightarrow q_n \cdot \chi_n$$

4. All irreversible channels are non-unary, or else the destinations of these channels should be sinks, and χ_n cannot be 0 length, since q is the destination of non-unary channel c.
5. A jumping circuit is obtained by using the channel c to complete the circuit.

Decidability

Using the above three lemmas, we can prove the decidability part of the main theorem.

1. First, split all irreversible unary channels to get a divided topology
2. In a divided topology, by lemma 3, all non-unary channels are essential.
3. We can synchronise each non-unary channel one by one to get a topology consisting only of unary channels, which is decidable.

Thank You!

THANK
YOU
GRACIAS
ARIGATO
SHUKURIA
DANKSCHEEN
BİYAN
SHUKRIA
TINGKI
YAQHANYELAY
TASHAKKUR ATU
SUKSAMA
EKHMET
GRAZIE
MEHRBANI
MAAKE
KOMAPSUMNIDA
PALDIES
GOZAIMASHITA
EFCHARISTO
JUSPAXAR
BOLZİN
MERCI