Communicating systems

Anca Muscholl

LaBRI, Bordeaux

Cremona, October'08

Asynchronously communicating systems

A basic notion

- distributed, multi-peer systems
- network protocols
- telecommunication systems
- ITU norms Z.120 (MSC, Message Sequence Charts), Z.100 (SDL, Specification and Description Language)

Asynchronously communicating systems

A basic notion

- distributed, multi-peer systems
- network protocols
- telecommunication systems
- ITU norms Z.120 (MSC, Message Sequence Charts), Z.100 (SDL, Specification and Description Language)

This talk: channel systems

- different approaches for verification (model-checking)
- language-theoretical framework for asynchronous communication (synthesis)

Model-checking and synthesis

Problems

- Model-checking
  1. Build an abstract model from a protocol/program
  2. Develop algorithmic verification techniques (exact/approximated)
- Synthesis (Realizability)
  1. Build a specification for a protocol/program (desirable/undesirable properties)
  2. Develop algorithms for constructing systems compliant with given specification (closed/open systems)

Model-checking and synthesis

Problems

- Model-checking
  1. Build an abstract model from a protocol/program
  2. Develop algorithmic verification techniques (exact/approximated)
- Synthesis (Realizability)
  1. Build a specification for a protocol/program (desirable/undesirable properties)
  2. Develop algorithms for constructing systems compliant with given specification (closed/open systems)

Challenges for communicating systems

- infinite-state systems: unbounded memory (channels)
- distributed peers (concurrency)

Communicating Finite State Machines, CFSM

- finite set $\mathcal{P}$ of processes (peers) $P, Q, \ldots$
- set Ch of unbounded channels (FIFO queues): $\mathsf{Ch} \subseteq \mathcal{P}^2 \setminus \mathsf{id}_{\mathcal{P}}$
- contents of channels: words over (finite) message alphabet C
- each process P: finite-state machine $\mathcal{A}_P = (S_P, \Sigma_P, \longrightarrow_P, s_P^0)$ with possible actions from Σ_P:
  - $P!Q(c)$: process P sends message c to process Q
  - $P?Q(c)$: process P receives message c from process Q
  - $P(\ell)$: process P performs local action ℓ

Communicating Finite State Machines, CFSM

- finite set $\mathcal{P}$ of processes (peers) $P, Q, \ldots$
- set Ch of unbounded channels (FIFO queues): $\mathsf{Ch} \subseteq \mathcal{P}^2 \setminus \mathsf{id}_{\mathcal{P}}$
- contents of channels: words over (finite) message alphabet C
- each process P: finite-state machine $\mathcal{A}_P = (S_P, \Sigma_P, \longrightarrow_P, s_P^0)$ with possible actions from Σ_P:
  - $P!Q(c)$: process P sends message c to process Q
  - $P?Q(c)$: process P receives message c from process Q
  - $P(\ell)$: process P performs local action ℓ

Configurations

- Configuration $\langle (s_P)_{P \in \mathcal{P}}, (W_{P,Q})_{(P,Q) \in \mathsf{Ch}} \rangle$:
  - local control states s_P, $P \in \mathcal{P}$
  - channel contents $W_{P,Q} \in C^*$, $(P,Q) \in \mathsf{Ch}$
- Transitions $\langle s, W \rangle \xrightarrow{a} \langle s', W' \rangle$, with $a \in \Sigma_P$:
  - $s_P \xrightarrow{a}_P s'_P$
  - if $a = P!Q(c)$ then $W'_{P,Q} = W_{P,Q}c$
  - if $a = P?Q(c)$ and $W_{P,Q} = cZ$ then $W'_{P,Q} = Z$

  All other components are unchanged.

Model-checking CFSM

Questions. Given a CFSM decide:

- Termination: are all computations from the initial configuration finite?
- Structural termination: are all computations (from any configuration) finite?
- Boundedness: is there some (uniform) bound $n > 0$ such that every reachable configuration is of size at most n?
- Reachability (EF s): is a given control state s (or configuration) reachable?
- Eventuality (AF s): do all computations eventually reach a given control state s?
- More general properties (CTL, LTL, CTL*)...

Model-checking CFSM

Questions. Given a CFSM decide:

- Termination: are all computations from the initial configuration finite?
- Structural termination: are all computations (from any configuration) finite?
- Boundedness: is there some (uniform) bound $n > 0$ such that every reachable configuration is of size at most n?
- Reachability (EF s): is a given control state s (or configuration) reachable?
- Eventuality (AF s): do all computations eventually reach a given control state s?
- More general properties (CTL, LTL, CTL*)...

Brand/Zafiropulo '82

CFSM are Turing-equivalent, hence all questions above are undecidable.

Model-checking CFSM

Questions. Given a CFSM decide:

- Termination: are all computations from the initial configuration finite?
- Structural termination: are all computations (from any configuration) finite?
- Boundedness: is there some (uniform) bound $n > 0$ such that every reachable configuration is of size at most n?
- Reachability (EF s): is a given control state s (or configuration) reachable?
- Eventuality (AF s): do all computations eventually reach a given control state s?
- More general properties (CTL, LTL, CTL*)...

Brand/Zafiropulo '82

CFSM are Turing-equivalent, hence all questions above are undecidable.

Techniques for communicating systems

- structural restrictions
- symbolic representations and approximating reachability sets
- faulty machines
- partial-order methods

Structural restrictions

Restricted channels

- consider only configurations of bounded size (given a bound n, channels are considered up to size n):
  CFSM become finite-state machines
- allow only one message type per channel ($|C| = 1$):
  CFSM are Petri nets

Limitations

- Both approximations are too coarse.
- None of the approaches avoids high complexity.

Techniques for communicating systems

- structural restrictions
- symbolic representations and approximating reachability sets
- faulty machines
- partial-order methods

Channel systems

- In the following, CFSM consist of set S of global control states + k FIFO channels. Configurations $\langle s, w_1, \ldots, w_k \rangle$ with $s \in S$ and $w_i \in \Sigma_i^*$.
- Let

$$\text{post}^*(s, C) = \{(s', C') \mid (s, C) \xrightarrow{*} (s', C')\}$$

  where C is a set of channel contents and s a control state.
- Symbolic representation of (s, C): finite description of

$$\{s\#w_1\#w_2\#\cdots w_k \mid \langle s, w_1, \ldots, w_k \rangle \in C\}$$

Symbolic representations

Pachl '82

- CFSM with recognizable channel property: $\text{post}^*(s, C)$ is a recognizable subset of $S \times \Sigma_1^* \times \cdots \times \Sigma_k^*$ (i.e., finite union of products $L_1 \times \cdots \times L_k$ with every L_i regular).
- Very restrictive (a process P sending in alternation to Q and R generates a non-recognizable reachability set).
- Shows decidability of deadlock-freedom for CFSM with recognizable channels.
- Non-constructive proof, searches for recognizable and transition-closed sets of configurations that witness non-reachability.
- CFSM with rational channel property: $\text{post}^*(s, C)$ is accepted by a finite k-tape automaton $\mathcal{A}_s$.
- For cyclic CFSM (1-way ring topology), recognizable and rational coincide.

Symbolic representations

Pachl '82

- CFSM with recognizable channel property: $\text{post}^*(s, C)$ is a recognizable subset of $S \times \Sigma_1^* \times \cdots \times \Sigma_k^*$ (i.e., finite union of products $L_1 \times \cdots \times L_k$ with every L_i regular).
- Very restrictive (a process P sending in alternation to Q and R generates a non-recognizable reachability set).
- Shows decidability of deadlock-freedom for CFSM with recognizable channels.
- Non-constructive proof, searches for recognizable and transition-closed sets of configurations that witness non-reachability.
- CFSM with rational channel property: $\text{post}^*(s, C)$ is accepted by a finite k-tape automaton $\mathcal{A}_s$.
- For cyclic CFSM (1-way ring topology), recognizable and rational coincide.

Peng, Purushothaman '92

- Reachability sets of cyclic, one-message-type CFSM are (effectively) semi-linear. (Shows that channel contents have context-free representations.)
- Polynomial-time algorithms for reachability, deadlock-freedom and boundedness (polynomial in the size of global state space.)

Approximating the reachability set

- Given a set of configurations $X \subseteq S \times \Sigma_1^* \times \cdots \times \Sigma_k^*$, the fix-point computation

$$X = X \cup \mathsf{post}(X)$$

  does not terminate in general.
- Control loops τ on state s: sequence of transitions starting and ending in s.
- Let

$$\mathsf{post}_\tau^*(s, C) = \{C' \mid (s, C) \xrightarrow{\tau^*} (s, C')\}$$

  where C is a set of channel contents, s a state, τ a control loop on s.
- Given a set of (control) loops Θ, compute

$$X = X \cup \mathsf{post}(X) \cup \bigcup_{\tau \in \Theta} \mathsf{post}_\tau^*(X)$$

  (Acceleration that might lead to termination.)

Symbolic representations

Use symbolic representations for channel contents

Given state s, a control loop τ on s and a set C of channel contents, compute the representation of $post^*_\tau(s, C)$.

Symbolic representations

Use symbolic representations for channel contents

Given state s, a control loop τ on s and a set C of channel contents, compute the representation of $post^*_\tau(s, C)$.

Queue-content decision diagrams, QDD

Boigelot/Godefroid '96, B/G/Willems/Wolper '97:

- QDD are finite automata accepting words of the form $w_1 \cdot w_2 \cdot \ldots w_k$.
- QDD correspond to recognizable channels.
- Control loops τ s.t. $post^*_\tau$ preserves QDDs are precisely those that "count" on at most one channel (e.g. $(P!Q\ Q?R)^*$, but not $(P!Q\ Q!R)^*$).

Symbolic representations

Use symbolic representations for channel contents

Given state s, a control loop τ on s and a set C of channel contents, compute the representation of $post^*_\tau(s, C)$.

Queue-content decision diagrams, QDD

Boigelot/Godefroid '96, B/G/Willems/Wolper '97:

- QDD are finite automata accepting words of the form $w_1 \cdot w_2 \cdot \ldots w_k$.
- QDD correspond to recognizable channels.
- Control loops τ s.t. $post^*_\tau$ preserves QDDs are precisely those that "count" on at most one channel (e.g. $(P!Q\ Q?R)^*$, but not $(P!Q\ Q!R)^*$).

CQDD = QDD + linear constraints

Bouajjani/Habermehl '97:

- Add linear constraints on the number of transitions of the finite automaton: CQDD (constrained queue decision diagrams).
- $post^*_\tau$ preserves CQDDs, for arbitrary control loops τ.
- CQDDs closed under various operations (but not complement).
  Inclusion of CQDDs is decidable for a restricted class of deterministic automata.

Techniques for communicating systems

- structural restrictions
- symbolic representations and approximating reachability sets
- faulty machines
- partial-order methods

Faulty CFSM

Faults

- Deletion errors (lossy machines): channels can loose messages (anywhere, anytime)
- Insertion errors: channels can insert messages (anywhere, anytime)

Faulty CFSM

Faults

- Deletion errors (lossy machines): channels can loose messages (anywhere, anytime)
- Insertion errors: channels can insert messages (anywhere, anytime)

Basics

- Subword ordering $\preceq$ ($abca \preceq aacbcbab$)
- A language $L \subseteq \Sigma^*$ is upward-closed if for every $v \in L$, $v \preceq w$: $w \in L$, too.

Faulty CFSM

Faults

- Deletion errors (lossy machines): channels can loose messages (anywhere, anytime)
- Insertion errors: channels can insert messages (anywhere, anytime)

Basics

- Subword ordering $\preceq$ ($abca \preceq aacbcbab$)
- A language $L \subseteq \Sigma^*$ is upward-closed if for every $v \in L$, $v \preceq w$: $w \in L$, too.

Higman's Lemma

Anti-chains w.r.t. subword ordering are finite.

Faulty CFSM

Faults

- Deletion errors (lossy machines): channels can loose messages (anywhere, anytime)
- Insertion errors: channels can insert messages (anywhere, anytime)

Basics

- Subword ordering $\preceq$ ($abca \preceq aacbcbab$)
- A language $L \subseteq \Sigma^*$ is upward-closed if for every $v \in L$, $v \preceq w$: $w \in L$, too.

Higman's Lemma

Anti-chains w.r.t. subword ordering are finite.

Consequences of Higman:

- Every upward-closed language is regular.
- Every strictly $\subseteq$-monotone sequence of upward-closed languages is finite.

Faulty CFSM

Faults

- Deletion errors (lossy machines): channels can loose messages (anywhere, anytime)
- Insertion errors: channels can insert messages (anywhere, anytime)

Basics

- Subword ordering $\preceq$ ($abca \preceq aacbcbab$)
- A language $L \subseteq \Sigma^*$ is upward-closed if for every $v \in L$, $v \preceq w$: $w \in L$, too.

Higman's Lemma

Anti-chains w.r.t. subword ordering are finite.

Consequences of Higman:

- Every upward-closed language is regular.
- Every strictly $\subseteq$-monotone sequence of upward-closed languages is finite.
- The reachability set of CFSM with insertion errors is regular and effectively computable. N.B.: the reachability set of a lossy CFSM is also regular (complement is upward-closed), although not effectively computable: boundedness is undecidable.

Well-structured transition systems

Abdulla/Jonsson, Finkel/Schnoebelen

- Infinite set S of states
- Decidable, well-founded preorder $\preceq$ on S with no infinite anti-chain.
- Monotone transitions:

$$\textit{If } s \xrightarrow{a} s' \textit{ and } s \preceq t, \textit{ then } t \xrightarrow{a} t' \textit{ for some } s' \preceq t'.$$

Well-structured transition systems

Abdulla/Jonsson, Finkel/Schnoebelen

- Infinite set S of states
- Decidable, well-founded preorder $\preceq$ on S with no infinite anti-chain.
- Monotone transitions:

  If $s \xrightarrow{a} s'$ *and* $s \preceq t$, *then* $t \xrightarrow{a} t'$ *for some* $s' \preceq t'$.

Forward analysis

The finite reachability tree (FRT) from a configuration C is a finite prefix of the unfolding, obtained by stopping at a node C' if there is some $C'' \preceq C'$ with $C \xrightarrow{*} C'' \xrightarrow{+} C'$. (Finiteness ensured by König's Lemma and Higman.)

Well-structured transition systems

Abdulla/Jonsson, Finkel/Schnoebelen

- Infinite set S of states
- Decidable, well-founded preorder $\preceq$ on S with no infinite anti-chain.
- Monotone transitions:

  If $s \xrightarrow{a} s'$ *and* $s \preceq t$, *then* $t \xrightarrow{a} t'$ *for some* $s' \preceq t'$.

Forward analysis

The finite reachability tree (FRT) from a configuration C is a finite prefix of the unfolding, obtained by stopping at a node C' if there is some $C'' \preceq C'$ with $C \xrightarrow{*} C'' \xrightarrow{+} C'$. (Finiteness ensured by König's Lemma and Higman.)

Backward analysis

- For a set of states T, denote by $\mathrm{pre}^*(T)$ the set $\{t' \mid t' \xrightarrow{*} t \text{ for some } t \in T\}$.
- If T is upward-closed, then $\mathrm{pre}^*(T)$ is upward-closed as well.
- Compute $\mathrm{pre}^*(T)$ for upward-closed T as least fix-point, by calculating at each step the set of minimal elements.

Lossy CFSM

- Instance of well-structured transition systems (w.r.t. subword ordering).
- Decidable questions: safety properties such as reachability, termination; eventuality property $AF\, s$.
  Methods: backward analysis (reachability), FRT (termination).
- Undecidable questions: boundedness, structural termination, liveness properties (visiting a control state infinitely often), CTL/LTL.
- Complexity for decidable questions: non-primitive recursive (Schnoebelen) - more precise characterization via the Extended Grzegorczyk Hierarchy.

Techniques for communicating systems

- structural restrictions
- symbolic representations and approximating reachability sets
- faulty machines
- partial-order methods

Model-checking error-free, unbounded CFSM

Language of CFSM

- CFSM $\mathcal{A}$: labeled (infinite) transition system, with label alphabet

$$\Sigma = \cup_P \Sigma_P = \{P!Q(c), P?Q(c) \mid c \in C\}$$

- Language $L(\mathcal{A}) \subseteq \Sigma^*$: words labeling accepting (finite) computations
- Conversation languages/protocols à la Bultan/Fu/Su.

Basic verification problems

- Model-checking: given a specification $S \subseteq \Sigma^*$ (e.g. LTL) and a CFSM $\mathcal{A}$, check if $L(\mathcal{A}) \cap S \neq \emptyset$.
- Realizability (synthesis): given a specification $S \subseteq \Sigma^*$, compute a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.

Model-checking error-free, unbounded CFSM

Language of CFSM

- CFSM $\mathcal{A}$: labeled (infinite) transition system, with label alphabet

$$\Sigma = \cup_P \Sigma_P = \{P!Q(c), P?Q(c) \mid c \in C\}$$

- Language $L(\mathcal{A}) \subseteq \Sigma^*$: words labeling accepting (finite) computations
- Conversation languages/protocols à la Bultan/Fu/Su.

Basic verification problems

- Model-checking: given a specification $S \subseteq \Sigma^*$ (e.g. LTL) and a CFSM $\mathcal{A}$, check if $L(\mathcal{A}) \cap S \neq \emptyset$.
- Realizability (synthesis): given a specification $S \subseteq \Sigma^*$, compute a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.

Which specifications?

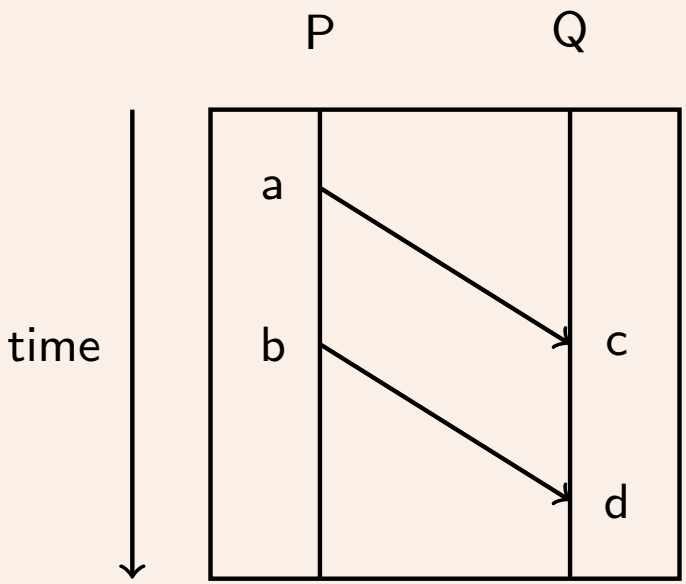
- Model-checking CFSM versus LTL is undecidable.
- LTL not useful as a specification formalism (example: Producer-consumer).
- Use extended LTL: new next-operator X_{msg}.
  $w, i \models X_{\mathsf{msg}}\varphi$ if w_i send event, w_j $(j > i)$ matching receive, and $w, j \models \varphi$.
  But: realizability for extended LTL is undecidable.

Specifying asynchronous communication

Requirements

- Explicit description of messages and of local concurrency.
- Sequential global control.

Partial order specifications: message sequence charts (MSC)



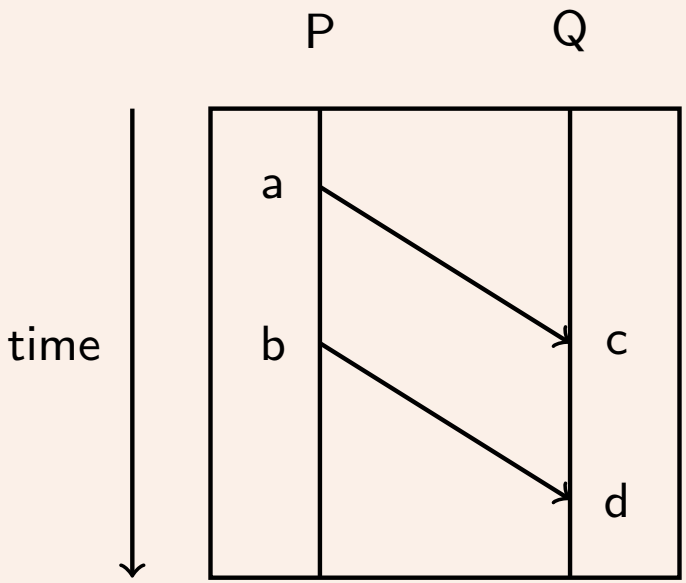
- events a, b, c, d
- processes P, Q
- messages $(a, c), (b, d)$
- $a <_P b$, $c <_Q d$
- partial order induced by messages + processes
- b, c unordered
- linearizations $acbd, abcd$ ($P!Q\, Q?P\, P!Q\, Q?P$ and $P!Q\, P!Q\, Q?P\, Q?P$).

Specifying asynchronous communication

Requirements

- Explicit description of messages and of local concurrency.
- Sequential global control.

Partial order specifications: message sequence charts (MSC)

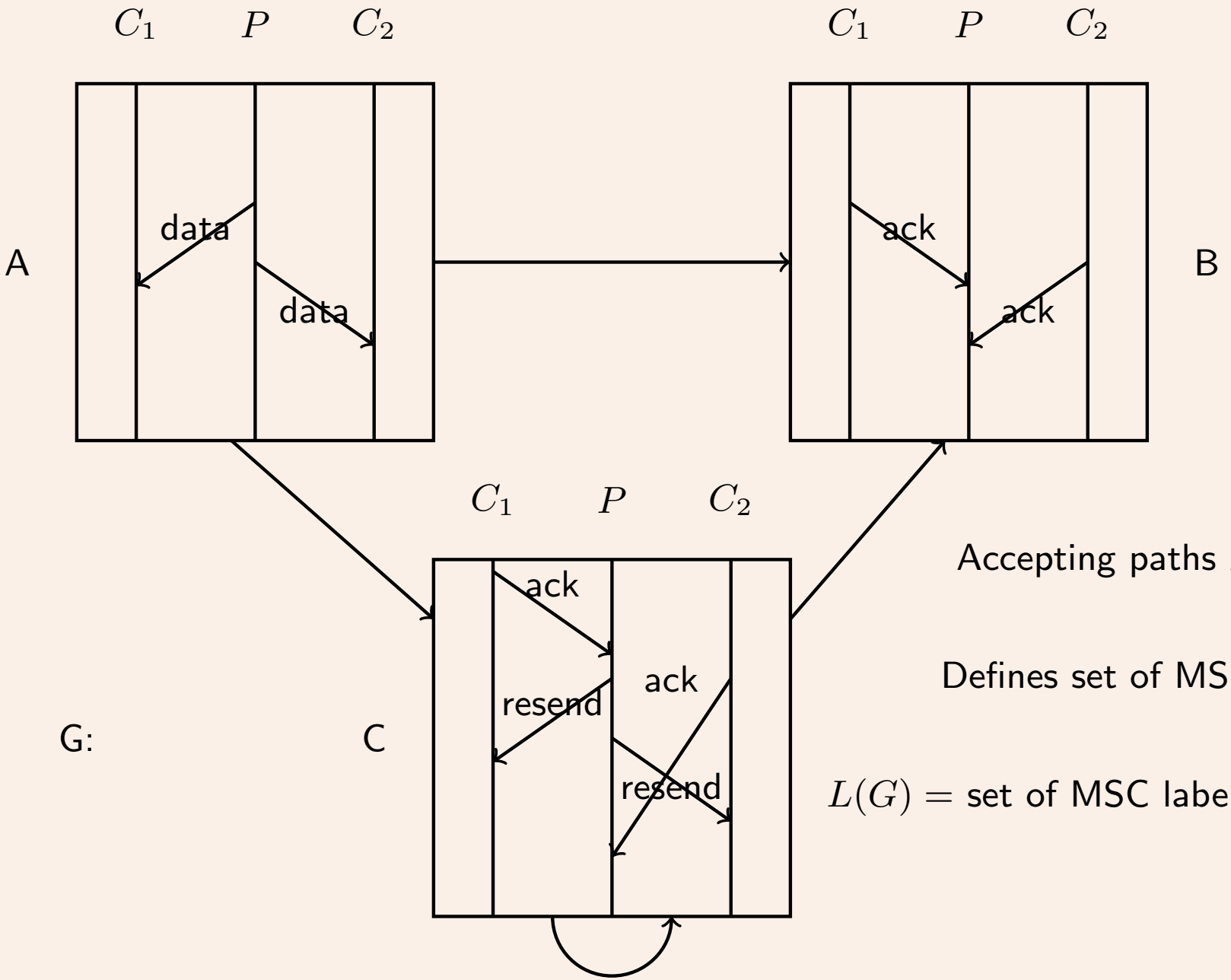


- events a, b, c, d
- processes P, Q
- messages $(a, c), (b, d)$
- $a <_P b$, $c <_Q d$
- partial order induced by messages + processes
- b, c unordered
- linearizations $acbd, abcd$ ($P!Q\, Q?P\, P!Q\, Q?P$ and $P!Q\, P!Q\, Q?P\, Q?P$).

CFSM and MSC

Each computation $w \in L(\mathcal{A})$ of CFSM $\mathcal{A}$ induces an MSC M_w.

MSC-graphs (ITU Z.120)



Accepting paths AB^*C

Defines set of MSC $L(G)$

$L(G) =$ set of MSC labeling acc. paths

MSC-graph specifications

Given a CFSM $\mathcal{A}$ and an MSC-graph specification S.

- Testing $L(\mathcal{A}) \cap L(S) \neq \emptyset$ is decidable in polynomial space.
- Testing $L(S) \subseteq L(\mathcal{A})$ is decidable in exponential space.

MSC-graph specifications

Given a CFSM $\mathcal{A}$ and an MSC-graph specification S.

- Testing $L(\mathcal{A}) \cap L(S) \neq \emptyset$ is decidable in polynomial space.
- Testing $L(S) \subseteq L(\mathcal{A})$ is decidable in exponential space.

Proof idea

- Representative linearizations $\mathcal{R}(S)$ of S: choose a linearization w_M of each MSC M labeling a node of S, and let

$$\mathcal{R}(S) = \{w_{M_1} w_{M_2} \cdots w_{M_k} \mid M_1 M_2 \cdots M_k \text{ accepting path in } S\}$$

MSC-graph specifications

Given a CFSM $\mathcal{A}$ and an MSC-graph specification S.

- Testing $L(\mathcal{A}) \cap L(S) \neq \emptyset$ is decidable in polynomial space.
- Testing $L(S) \subseteq L(\mathcal{A})$ is decidable in exponential space.

Proof idea

- Representative linearizations $\mathcal{R}(S)$ of S: choose a linearization w_M of each MSC M labeling a node of S, and let

$$\mathcal{R}(S) = \{w_{M_1} w_{M_2} \cdots w_{M_k} \mid M_1 M_2 \cdots M_k \text{ accepting path in } S\}$$

- $w \in \Sigma^*$ is B-bounded, if every $v \leq w$ satisfies $||w|_{P!Q} - |w|_{Q?P}| \leq B$ (for all P, Q).

MSC-graph specifications

Given a CFSM $\mathcal{A}$ and an MSC-graph specification S.

- Testing $L(\mathcal{A}) \cap L(S) \neq \emptyset$ is decidable in polynomial space.
- Testing $L(S) \subseteq L(\mathcal{A})$ is decidable in exponential space.

Proof idea

- Representative linearizations $\mathcal{R}(S)$ of S: choose a linearization w_M of each MSC M labeling a node of S, and let

$$\mathcal{R}(S) = \{w_{M_1} w_{M_2} \cdots w_{M_k} \mid M_1 M_2 \cdots M_k \text{ accepting path in } S\}$$

- $w \in \Sigma^*$ is B-bounded, if every $v \leq w$ satisfies $||w|_{P!Q} - |w|_{Q?P}| \leq B$ (for all P, Q).
- $\mathcal{R}(S) \subseteq \Sigma^*$ is regular and there is some $B_S > 0$ such that every $w \in \mathcal{R}(S)$ is B_S-bounded.

MSC-graph specifications

Given a CFSM $\mathcal{A}$ and an MSC-graph specification S.

- Testing $L(\mathcal{A}) \cap L(S) \neq \emptyset$ is decidable in polynomial space.
- Testing $L(S) \subseteq L(\mathcal{A})$ is decidable in exponential space.

Proof idea

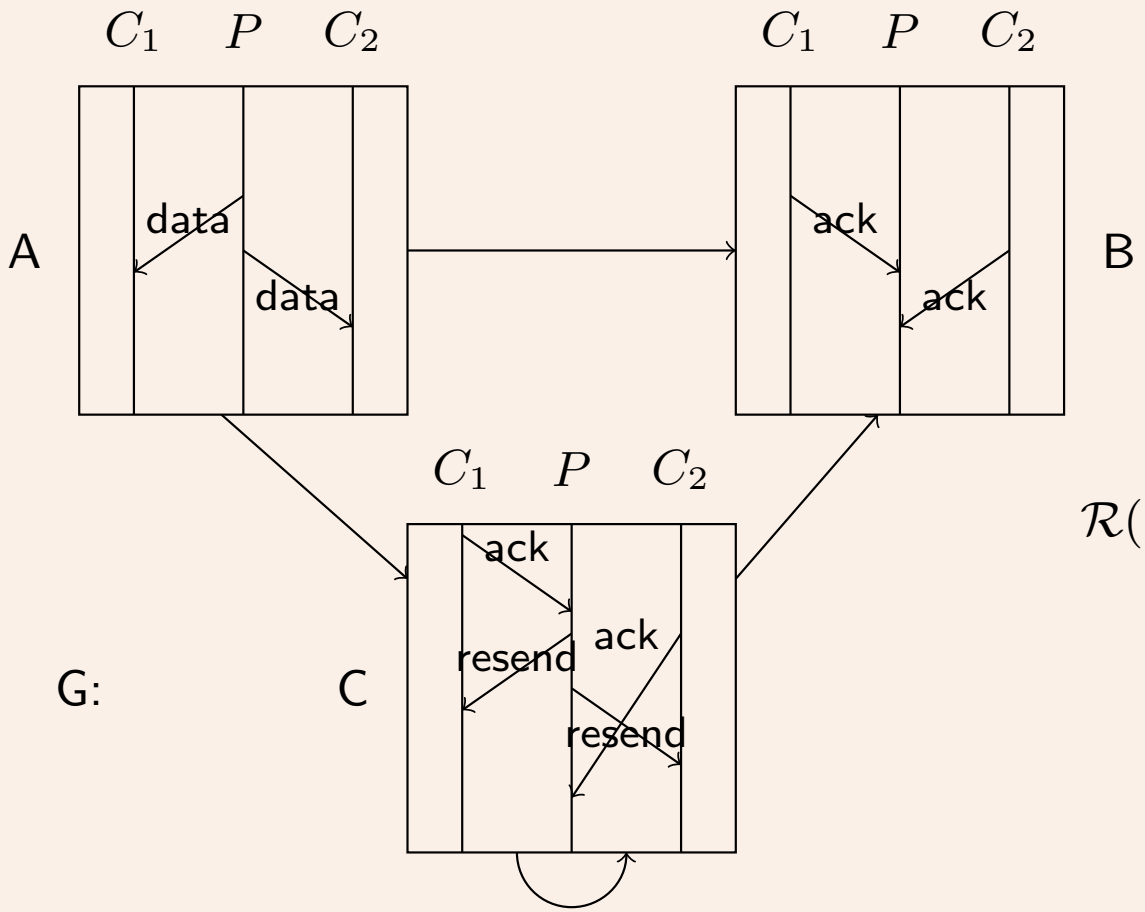
- Representative linearizations $\mathcal{R}(S)$ of S: choose a linearization w_M of each MSC M labeling a node of S, and let

$$\mathcal{R}(S) = \{w_{M_1} w_{M_2} \cdots w_{M_k} \mid M_1 M_2 \cdots M_k \text{ accepting path in } S\}$$

- $w \in \Sigma^*$ is B-bounded, if every $v \leq w$ satisfies $||w|_{P!Q} - |w|_{Q?P}| \leq B$ (for all P, Q).
- $\mathcal{R}(S) \subseteq \Sigma^*$ is regular and there is some $B_S > 0$ such that every $w \in \mathcal{R}(S)$ is B_S-bounded.
- Compute the finite-state restriction $\mathcal{A}_S$ of the CFSM $\mathcal{A}$, by considering only configurations with channels of size B_S.
- $L(\mathcal{A}) \cap L(S) \neq \emptyset$ iff $L(\mathcal{A}_S) \cap \mathcal{R}(S) \neq \emptyset$
- $L(S) \subseteq L(\mathcal{A})$ iff $\mathcal{R}(S) \subseteq L(\mathcal{A}_S)$

Example

Specification S



$\mathcal{R}(S)$ is 1-bounded

$w_A = P!C_1(\mathsf{data})C_1?P(\mathsf{data})P!C_2(\mathsf{data})C_2?P(\mathsf{data})$

$w_C =$

$C_1!P(\mathsf{ack})P?C_1(\mathsf{ack})P!C_1(\mathsf{rsnd})C_1?P(\mathsf{rsnd})P!C_2(\mathsf{rsnd})C_2!P(\mathsf{ack})P?C_2(\mathsf{ack})C_2?P(\mathsf{rsnd})$

Model-checking (cont.)

Complementing MSC-graph specifications

- Testing $L(\mathcal{A}) \subseteq L(S)$ for CFSM $\mathcal{A}$ and MSC-graph S: can we complement S?

Model-checking (cont.)

Complementing MSC-graph specifications

- Testing $L(\mathcal{A}) \subseteq L(S)$ for CFSM $\mathcal{A}$ and MSC-graph S: can we complement S?
- Yes: syntactic restriction on MSC-graphs (globally-cooperative).

Model-checking (cont.)

Complementing MSC-graph specifications

- Testing $L(\mathcal{A}) \subseteq L(S)$ for CFSM $\mathcal{A}$ and MSC-graph S: can we complement S?
- Yes: syntactic restriction on MSC-graphs (globally-cooperative).

Communication graph

Given an MSC M over process set $\mathcal{P}$, its communication graph G_M has vertex set $\mathcal{P}$ and edges (P, Q) if M contains at least one message from P to Q.

Model-checking (cont.)

Complementing MSC-graph specifications

- Testing $L(\mathcal{A}) \subseteq L(S)$ for CFSM $\mathcal{A}$ and MSC-graph S: can we complement S?
- Yes: syntactic restriction on MSC-graphs (globally-cooperative).

Communication graph

Given an MSC M over process set $\mathcal{P}$, its communication graph G_M has vertex set $\mathcal{P}$ and edges (P, Q) if M contains at least one message from P to Q.

Globally-cooperative MSC-graph

MSC-graph S is globally-cooperative if the communication graph of every loop is weakly connected.

Model-checking (cont.)

Complementing MSC-graph specifications

- Testing $L(\mathcal{A}) \subseteq L(S)$ for CFSM $\mathcal{A}$ and MSC-graph S: can we complement S?
- Yes: syntactic restriction on MSC-graphs (globally-cooperative).

Communication graph

Given an MSC M over process set $\mathcal{P}$, its communication graph G_M has vertex set $\mathcal{P}$ and edges (P, Q) if M contains at least one message from P to Q.

Globally-cooperative MSC-graph

MSC-graph S is globally-cooperative if the communication graph of every loop is weakly connected.

Model-checking

Genest/M/Seidl/Zeitoun 04:

- Given a globally-cooperative MSC-graph S, one can construct an exponential size automaton $\mathcal{B}$ that accepts all B_S-bounded linearizations of MSC in $L(S)$.
- Compute the finite-state restriction $\mathcal{A}_S$ of the CFSM $\mathcal{A}$, by considering only configurations with channels of size B_S.
- $L(\mathcal{A}) \subseteq L(S)$ iff $L(\mathcal{A}_S) \cap L(\mathcal{B})^{\text{co}} \neq \emptyset$.

Restricted channels

$w \in \Sigma^*$ is B-bounded, if every $v \leq w$ satisfies $||w|_{P!Q} - |w|_{Q?P}| \leq B$ (for all P, Q).

Universal channel bounds

- CFSM $\mathcal{A}$ is universally B-bounded if every (accepting) computation of $\mathcal{A}$ is B-bounded.
- CFSM is universally-bounded if it is universally B-bounded for some B.
- $\mathcal{A}$ is universally-bounded iff it is a finite transition system.

Restricted channels

$w \in \Sigma^*$ is B-bounded, if every $v \leq w$ satisfies $||w|_{P!Q} - |w|_{Q?P}| \leq B$ (for all P, Q).

Universal channel bounds

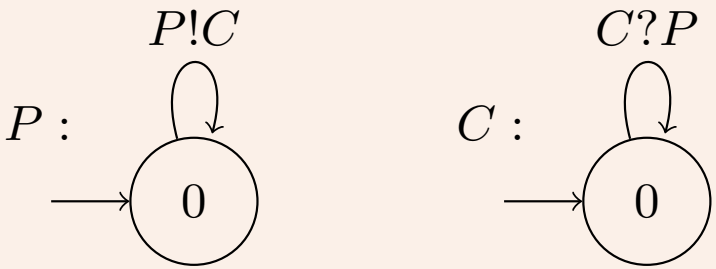
- CFSM $\mathcal{A}$ is universally B-bounded if every (accepting) computation of $\mathcal{A}$ is B-bounded.
- CFSM is universally-bounded if it is universally B-bounded for some B.
- $\mathcal{A}$ is universally-bounded iff it is a finite transition system.

Existential channel bounds

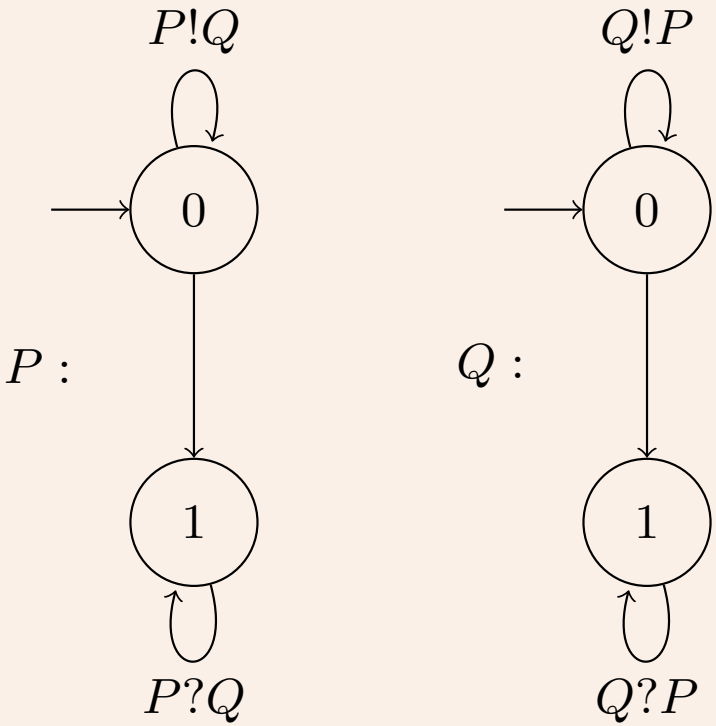
- CFSM $\mathcal{A}$ is existentially B-bounded if every (accepting) computation of $\mathcal{A}$ can be reordered in such a way that it becomes B-bounded.
- Example: $P!Q\ P!Q\ Q?P\ Q?P$ is 2-bounded, not 1-bounded.
  1-bounded reordering: $P!Q\ Q?P\ P!Q\ Q?P$.
- CFSM is existentially-bounded if it is existentially B-bounded for some B.
- Existential bound B: messages can be scheduled in such a way that there are never more than B pending messages on a given channel.

Channel bounds

Example



not universally-bounded,
existentially 1-bounded
(Producer-Consumer)



not existentially bounded

Deciding channel bounds

- It is undecidable to know whether $\mathcal{A}$ is universally (or existentially) bounded.
- For given B and assuming that $\mathcal{A}$ is deadlock-free, to know whether $\mathcal{A}$ is universally (existentially) B-bounded is decidable in polynomial space (Genest/Kuske/M. '07).

Model-checking and existentially-bounded CFSM

Let $\mathcal{A}$ be an existentially B-bounded CFSM and let $\mathcal{B}$ be an arbitrary CFSM. We can decide

- $L(\mathcal{A}) \cap L(\mathcal{B}) \neq \emptyset$ in polynomial space.
- $L(\mathcal{B}) \subseteq L(\mathcal{A})$ in exponential space.

Realizability

The problem

- Given a specification S (LTL, MSC-graph, MSO over partial orders), construct a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.

Realizability

The problem

- Given a specification S (LTL, MSC-graph, MSO over partial orders), construct a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.
- Closed & distributed synthesis (no environment, but distributed control).

Realizability

The problem

- Given a specification S (LTL, MSC-graph, MSO over partial orders), construct a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.
- Closed & distributed synthesis (no environment, but distributed control).
- Our approach: piggy-back additional control data in the messages allowed by S.

Realizability

The problem

- Given a specification S (LTL, MSC-graph, MSO over partial orders), construct a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.
- Closed & distributed synthesis (no environment, but distributed control).
- Our approach: piggy-back additional control data in the messages allowed by S.

Solutions

The realizability problem can be solved for restricted specifications. The most general solution yields existentially-bounded CFSM.

Realizability

The problem

- Given a specification S (LTL, MSC-graph, MSO over partial orders), construct a CFSM $\mathcal{A}$ with $L(\mathcal{A}) = S$, if possible.
- Closed & distributed synthesis (no environment, but distributed control).
- Our approach: piggy-back additional control data in the messages allowed by S.

Solutions

The realizability problem can be solved for restricted specifications. The most general solution yields existentially-bounded CFSM.

First attempt: regular specifications

- A language $L \subseteq \Sigma^*$ is closed, if there exists a set of MSC X such that L is the set of linearizations of MSC in X.
- It can be checked in polynomial space whether a regular language is closed.

Realizability for regular specifications

Universally bounded CFSM

(Mukund et al. '00, Kuske '02)
Let $L \subseteq \Sigma^*$ be regular. The following are equivalent characterizations:

- L is closed.
- $L = L(\mathcal{A})$ for some (universally bounded) CFSM.
- $L = L(\mathcal{A})$ for some (universally bounded) deterministic CFSM.

Some more characterizations (Kleene theorem)

(Mukund et al. '00, Kuske '02)
Over universally bounded MSC sets:

- MSO($\leq$), CFSM and MSC-graphs with strongly connected communication loops have the same expressive power.
- the logics MSO($\leq$), EMSO($\leq$), and EMSO($(\text{succ}_P)_{P\in\mathcal{P}}$, msg) are equally expressive.

Beyond finite-state case: existentially-bounded CFSM

B-closed specifications

- Example: $(P!C\ C?P)^*$ is regular, not closed, but realizable as CFSM.
- A language $L \subseteq \Sigma^*$ is B-closed if L is the set of B-bounded linearizations of some set X of MSC.
- It can be checked in polynomial space whether a regular language is B-closed, for given B.

Beyond finite-state case: existentially-bounded CFSM

B-closed specifications

- Example: $(P!C\ C?P)^*$ is regular, not closed, but realizable as CFSM.
- A language $L \subseteq \Sigma^*$ is B-closed if L is the set of B-bounded linearizations of some set X of MSC.
- It can be checked in polynomial space whether a regular language is B-closed, for given B.

Existentially bounded case

(Genest, Kuske, M. '04)
Let $L \subseteq \Sigma^*$ be regular. The following are equivalent characterizations:

- L is B-closed.
- L is the set of B-bounded computations of some existentially B-bounded CFSM.
- The set of MSC having some linearization in L is definable in one of the equivalent logics MSO($\leq$), EMSO($\leq$), and EMSO($(\mathrm{succ}_P)_{P\in\mathcal{P}}$, msg).

Some remarks on the proof

- The hard part is the construction of the CFSM from the regular language L (both universal and existential case) - this is distributed (closed) synthesis.
- Essential proof ingredient: Zielonka's construction of asynchronous automata (distributed automata with shared memory) & some more Mazurkiewicz trace theory.
- Main technical difficulty in the existential case: construct CFSM that deadlocks on computations that do not have a B-bounded reordering.

Conclusions

Two orthogonal approaches:

- channel systems handle messages explicitly (symbolic representations)
- language approach (MSC) handles messages implicitly

Various methods:

- error-free channel systems: under-approximation of reachability sets
- lossy channel systems: only simple properties are decidable, high complexity
- error-free CFSM: model-checking is feasible w.r.t. the right specifications (partially-ordered); synthesis is restricted to CFSM with existentially-bounded channels