Safety Verification of Communicating One-Counter Machines

A. Heußner — Univ. of Bamberg
T. Le Gall — CEA Saclay
G. Sutre — LaBRI Bordeaux

FSTTCS 2012, Hyderabad

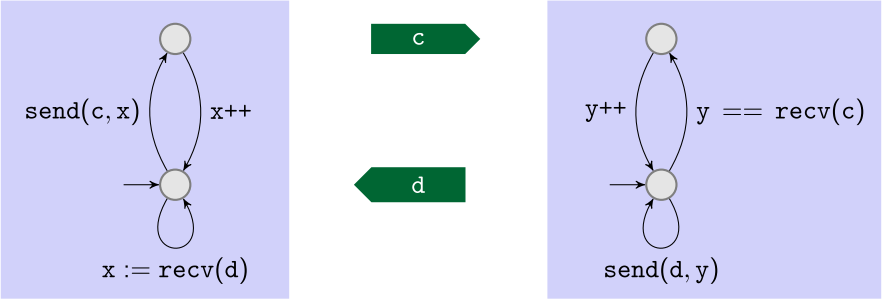
Agenda

- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of “Only If” Direction
- ➪ Main Theorem: Proof of “If” Direction
- ➪ Related/On-going/Future Work

Agenda

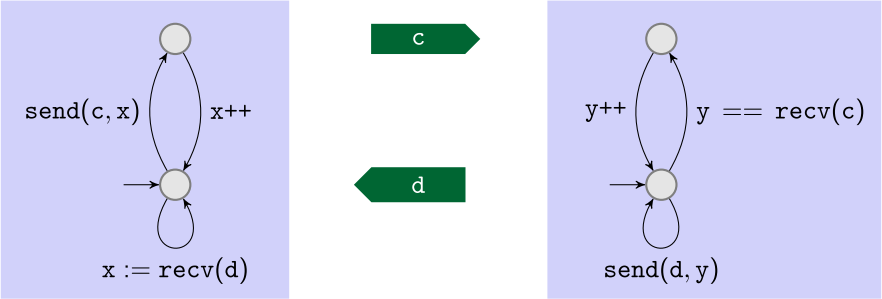
- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of “Only If” Direction
- ➪ Main Theorem: Proof of “If” Direction
- ➪ Related/On-going/Future Work

Example Sliding Window Protocol



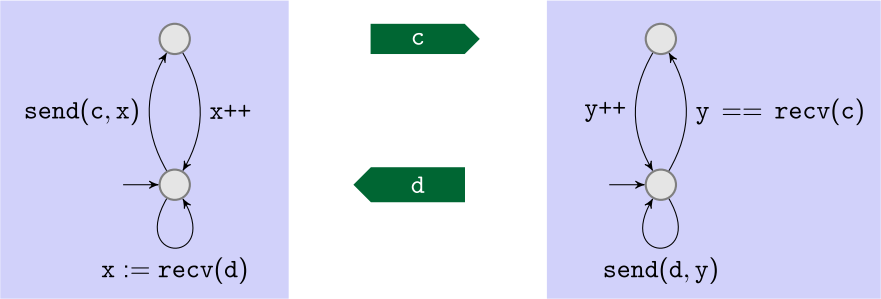
- One local counter for each process (x, y)

Example Sliding Window Protocol



- ➪ One local counter for each process (x, y)
- ➪ Asynchronous communication via perfect channels (c, d)
  - send the counter's value
  - receive and test/overwrite the counter

Example Sliding Window Protocol



- One local counter for each process (x, y)
- Asynchronous communication via perfect channels (c, d)
  - send the counter's value
  - receive and test/overwrite the counter
- Sources of infinity: local counters, message alphabet, channel length

Safety Verification of Example

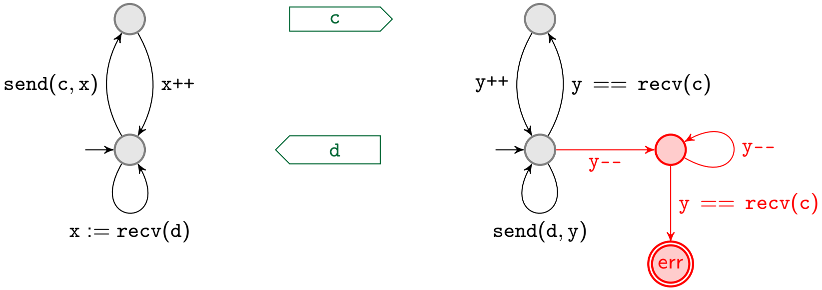
Goal:

Check absence of unspecified receptions due to y being too large

Safety Verification of Example

Goal:

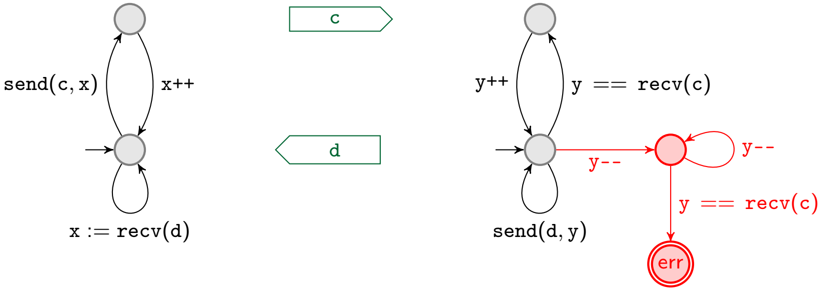
Check absence of unspecified receptions due to y being too large



Safety Verification of Example

Goal:

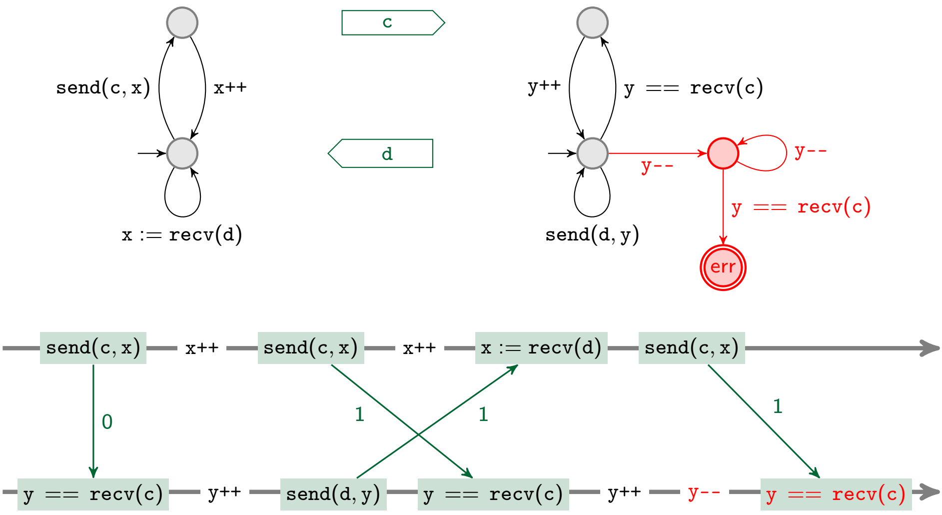
Check absence of unspecified receptions due to y being too large



More formal goal:

Check that err is not reachable

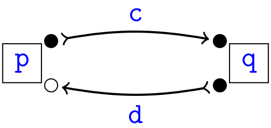
An Erroneous Execution



Agenda

- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of “Only If” Direction
- ➪ Main Theorem: Proof of “If” Direction
- ➪ Related/On-going/Future Work

Communication Topologies



Definition:

A topology is $\mathcal{T} = \langle P, C, \mathrm{src}, \mathrm{dst} \rangle$ where

- ➪ P : finite set of processes
- ➪ C : finite set of channels
- ➪ $\mathrm{src}, \mathrm{dst} : C \to P \times \{\bullet, \circ\}$

Communication types

- $\bullet$ strong : standard CFSM-style communication ($==$)
- $\circ$ weak : counter lost by communication ($:=$)

Communicating One-Counter Machines

Definition:

A communicating one-counter machine is $\langle S, I, F, A, \Delta \rangle$ where

- S : finite set of states
- $I, F \subseteq S$: initial and final states
- A : finite set of actions
- $\Delta \subseteq S \times A \times S$: finite set of transition rules

Communicating One-Counter Machines

Definition:

A communicating one-counter machine is $\langle S, I, F, A, \Delta \rangle$ where

- S : finite set of states
- $I, F \subseteq S$: initial and final states
- A : finite set of actions
- $\Delta \subseteq S \times A \times S$: finite set of transition rules

Actions : $\quad \texttt{add}(k) \mid \texttt{test}(\varphi) \mid c\,! \mid c\,? \qquad (k \in \mathbb{Z}, \varphi \in Presb_1, c \in C)$

Communicating One-Counter Machines

Definition:

A communicating one-counter machine is $\langle S, I, F, A, \Delta \rangle$ where

- S : finite set of states
- $I, F \subseteq S$: initial and final states
- A : finite set of actions
- $\Delta \subseteq S \times A \times S$: finite set of transition rules

Actions : $\quad \mathtt{add}(k) \mid \mathtt{test}(\varphi) \mid c\,! \mid c\,? \qquad (k \in \mathbb{Z}, \varphi \in \mathit{Presb}_1, c \in C)$

Definition:

A system of comm. one-counter machines is $\langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ where

- $\mathcal{T}$: topology
- $\mathcal{M}^p$: communicating one-counter machine

SC1CM Semantics: Configurations

Recall:

A SC1CM is $\langle \mathcal{T}, (\mathcal{M}^p)_{p\in P} \rangle$ where $\mathcal{M}^p = \langle S^p, I^p, F^p, A^p, \Delta^p \rangle$

$$\text{A configuration is} \quad \big(\underset{\textstyle s}{\overset{\prod_{p\in P} S^p}{\Cup}} \; , \; \underset{\textstyle x}{\overset{\mathbb{N}^P}{\Cup}} \; , \; \underset{\textstyle w}{\overset{(\mathbb{N}^*)^C}{\Cup}} \big)$$

$$\begin{array}{rcl} \textit{initial} & \overset{\text{def}}{\Longleftrightarrow} & s^p \in I^p \\ \textit{final} & \overset{\text{def}}{\Longleftrightarrow} & s^p \in F^p \end{array} \quad \wedge \quad x = \mathbf{0} \quad \wedge \quad w = \varepsilon$$

SC1CM Semantics: Transitions

Recall:

A SC1CM is $\langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ where $\mathcal{M}^p = \langle S^p, I^p, F^p, A^p, \Delta^p \rangle$

The transition relation $(s, x, w) \xrightarrow{a} (s', x', w')$ is defined by

- exactly one process moves
- counter actions behave as expected

SC1CM Semantics: Transitions

Recall:

A SC1CM is $\langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ where $\mathcal{M}^p = \langle S^p, I^p, F^p, A^p, \Delta^p \rangle$

The transition relation $(s, x, w) \xrightarrow{a} (s', x', w')$ is defined by

- exactly one process moves
- counter actions behave as expected
- communication actions depend on the endpoint's type

$$
\begin{array}{llllllll}
\bullet\!\!\xrightarrow{c} & c\,! & \equiv & c\,!\,x & \qquad \circ\!\!\xrightarrow{c} & c\,! & \equiv & c\,!\,x \;;\; x := \mathit{any} \\
\xrightarrow{c}\!\!\bullet & c\,? & \equiv & c\,?\,x & \qquad \xrightarrow{c}\!\!\circ & c\,? & \equiv & x := \mathit{any} \;;\; c\,?\,x
\end{array}
$$

SC1CM Semantics: Transitions

Recall:

A SC1CM is $\langle \mathcal{T}, (\mathcal{M}^p)_{p \in P} \rangle$ where $\mathcal{M}^p = \langle S^p, I^p, F^p, A^p, \Delta^p \rangle$

The transition relation $(s, x, w) \xrightarrow{a} (s', x', w')$ is defined by

- exactly one process moves
- counter actions behave as expected
- communication actions depend on the endpoint's type

$$\bullet\!\xrightarrow{c} \quad c\,! \quad \equiv \quad c\,!\,x \qquad\qquad \circ\!\xrightarrow{c} \quad c\,! \quad \equiv \quad c\,!\,x \;;\; x := any$$

$$\xrightarrow{c}\!\bullet \quad c\,? \quad \equiv \quad c\,?\,x \qquad\qquad \xrightarrow{c}\!\circ \quad c\,? \quad \equiv \quad x := any \;;\; c\,?\,x$$

- Note: $\circ$ can be simulated by $\bullet$

Parametrized Reachability Problem

Definition:

Given a topology $\mathcal{T}$, the decision problem $\textsc{Rp-Sc1cm}(\mathcal{T})$ is

Input: a system of communicating one-counter machines $\mathcal{S}$ with topology $\mathcal{T}$

Output: whether there exists a full run in $[\![\mathcal{S}]\!]$

A run $(s, x, w) \xrightarrow{*} (s', x', w')$ is full when $\begin{cases} (s, x, w) & \text{is initial} \\ (s', x', w') & \text{is final} \end{cases}$

Parametrized Reachability Problem

Definition:

Given a topology $\mathcal{T}$, the decision problem $\textsc{Rp-Sc1cm}(\mathcal{T})$ is

Input: a system of communicating one-counter machines $\mathcal{S}$ with topology $\mathcal{T}$

Output: whether there exists a full run in $[\![\mathcal{S}]\!]$

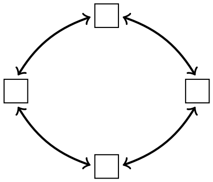
A run $(s, x, w) \xrightarrow{*} (s', x', w')$ is full when $\begin{cases} (s, x, w) & \text{is} \quad \text{initial} \\ (s', x', w') & \text{is} \quad \text{final} \end{cases}$

Goal:

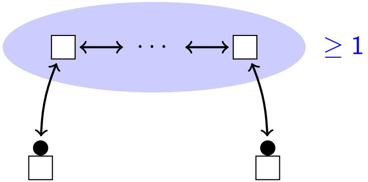
Characterize the topologies $\mathcal{T}$ where $\textsc{Rp-Sc1cm}(\mathcal{T})$ is decidable.

Main Result

Simple Undirected Cycle

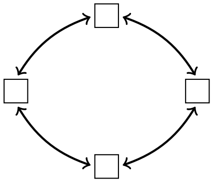


Simple Undirected Shunt

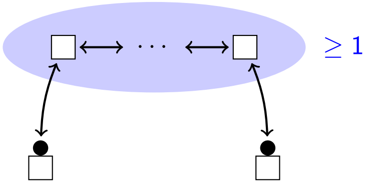


Main Result

Simple Undirected Cycle



Simple Undirected Shunt



Theorem:

$\textsc{Rp-Sc1cm}(\mathcal{T})$ is decidable iff $\mathcal{T}$ is cycle-free and shunt-free

- cycle-free: no simple undirected cycle
- shunt-free: no simple undirected shunt

Agenda

- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of "Only If" Direction
- ➪ Main Theorem: Proof of "If" Direction
- ➪ Related/On-going/Future Work

Reduce known Undecidability Results

Idea:

Simulation of communicating finite-state machines (CFSM)

- ➪ encode finite message exchange in exchange of counters
- ➪ reduce undecidability of reachability on cyclic architectures ↯

Reduce known Undecidability Results

Idea:

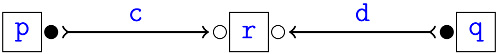
Simulation of communicating finite-state machines (CFSM)

➪ encode finite message exchange in exchange of counters

➪ reduce undecidability of reachability on cyclic architectures ↯

Idea:

Simulation of two-counters Minsky machines



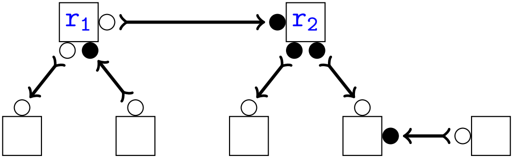
➪ reduce undecidability of reachability on simple shunt ↯

Agenda

- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of “Only If” Direction
- ➪ Main Theorem: Proof of “If” Direction
- ➪ Related/On-going/Future Work

Cycle-free & Shunt-free Topologies

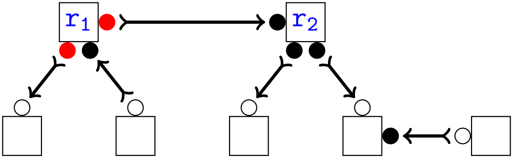
Form of topologies that are weakly-connected, cycle-free and shunt-free



- Two "roots" $\texttt{r}_1 \rightarrowtail \texttt{r}_2$
- Every simple undirected path from $\{\texttt{r}_1, \texttt{r}_2\}$ to $p \notin \{\texttt{r}_1, \texttt{r}_2\}$ ends with $\cdots \longleftrightarrow\!\circ\, p$

Cycle-free & Shunt-free Topologies

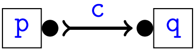
Form of topologies that are weakly-connected, cycle-free and shunt-free



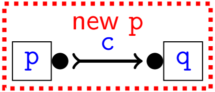
- ➪ Two "roots" $\mathtt{r}_1 \bullet\!\!\rightarrowtail\!\!\!\rightarrow\!\bullet\, \mathtt{r}_2$
- ➪ Every simple undirected path from $\{\mathtt{r}_1, \mathtt{r}_2\}$ to $p \notin \{\mathtt{r}_1, \mathtt{r}_2\}$ ends with $\cdots \bullet\!\longleftrightarrow\!\circ\, p$

[Recall: $\circ$ can be simulated by $\bullet$]

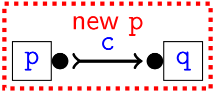
Case of Two Processes



Case of Two Processes



Case of Two Processes



Idea:

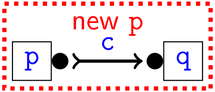
Intersect reachability relations of p and q between synchronizations

$$\cdots \xrightarrow{\mathtt{c}\,!\,\mathtt{c}\,?} \underbrace{(s, x) \xrightarrow{*} (t, y)}_{\chi_{s,t}} \xrightarrow{\mathtt{c}\,!\,\mathtt{c}\,?} \cdots$$

$(s^{\mathtt{p}}, s^{\mathtt{q}})$

$(t^{\mathtt{p}}, t^{\mathtt{q}})$

Case of Two Processes



Idea:

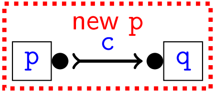
Intersect reachability relations of p and q between synchronizations

$$\cdots \xrightarrow{c\,!\,c\,?} \underbrace{(s, x) \xrightarrow{*} (t, y)}_{\chi_{s,t}} \xrightarrow{c\,!\,c\,?} \cdots$$

(s^{p}, s^{q}) (t^{p}, t^{q})

$$\chi_{s,t}(x, y) \;=\; (s^{p}, x) \xrightarrow{*}_{p} (t^{p}, y) \;\wedge\; (s^{q}, x) \xrightarrow{*}_{q} (t^{q}, y) \;\in\; Presb_2$$

Case of Two Processes



Idea:

Intersect reachability relations of $\mathtt{p}$ and $\mathtt{q}$ between synchronizations

$$\cdots \xrightarrow{\mathtt{c}\,!\,\mathtt{c}\,?} \underbrace{(s, x) \xrightarrow{\;*\;} (t, y)}_{\chi_{s,t}} \xrightarrow{\mathtt{c}\,!\,\mathtt{c}\,?} \cdots$$

$(s^{\mathtt{p}}, s^{\mathtt{q}})$ $(t^{\mathtt{p}}, t^{\mathtt{q}})$

$$\chi_{s,t}(x, y) \;=\; (s^{\mathtt{p}}, x) \xrightarrow{\;*\;}_{\mathtt{p}} (t^{\mathtt{p}}, y) \;\wedge\; (s^{\mathtt{q}}, x) \xrightarrow{\;*\;}_{\mathtt{q}} (t^{\mathtt{q}}, y) \;\in\; \mathit{Presb}_2$$

Issue:

Reachability is undecidable ↯ for the class of one-counter machines with Presburger-definable updates

One-Counter Reachability Relations

Fix two distinguished Presburger variables $\mathtt{x}$ and $\mathtt{y}$

The class of one-counter Presburger predicates is generated by

$$\psi \quad ::= \quad \varphi(\mathtt{x}) \mid \varphi(\mathtt{y}) \mid \varphi(\mathtt{x}-\mathtt{y}) \mid \varphi(\mathtt{y}-\mathtt{x}) \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{tt} \mid \mathtt{ff}$$

where φ ranges over unary Presburger predicates

One-Counter Reachability Relations

Fix two distinguished Presburger variables $\mathtt{x}$ and $\mathtt{y}$

The class of one-counter Presburger predicates is generated by

$$\psi \quad ::= \quad \varphi(\mathtt{x}) \mid \varphi(\mathtt{y}) \mid \varphi(\mathtt{x} - \mathtt{y}) \mid \varphi(\mathtt{y} - \mathtt{x}) \mid \psi \wedge \psi \mid \psi \vee \psi \mid \mathtt{tt} \mid \mathtt{ff}$$

where φ ranges over unary Presburger predicates

Theorem:

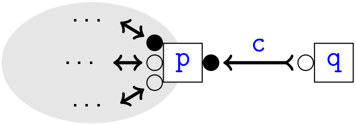
For every binary relation $R \subseteq \mathbb{N} \times \mathbb{N}$, the two following assertions are equivalent:

i) $R = \{(x, y) \mid (s, x) \xrightarrow{*} (t, y)\}$ for some one-counter machine

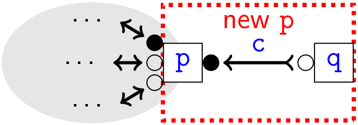
ii) $R = [\![\psi]\!]$ for some one-counter Presburger predicate ψ

➪ $\chi_{s,t}(x, y)$ can be translated into a one-counter machine ✓

Merging Leaf Processes



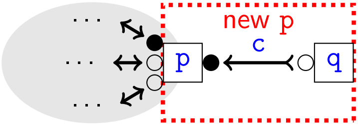
Merging Leaf Processes



Idea:

Merge leaf process q into p by summarizing q's behavior

Merging Leaf Processes

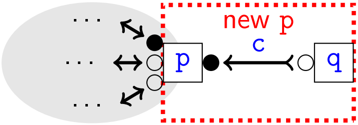


Idea:

Merge leaf process q into p by summarizing q's behavior

➪ Schedule q last : q moves only when p attempts to receive from c

Merging Leaf Processes

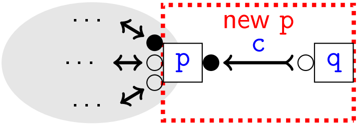


Idea:

Merge leaf process q into p by summarizing q's behavior

- ➪ Schedule q last : q moves only when p attempts to receive from c
- ➪ Communications between p and q become synchronizations $c\,!\cdot c\,?$

Merging Leaf Processes

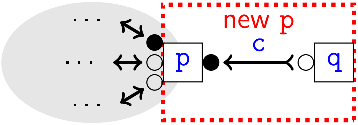


Idea:

Merge leaf process q into p by summarizing q's behavior

- Schedule q last : q moves only when p attempts to receive from c
- Communications between p and q become synchronizations $c\,!\cdot c\,?$
- States of p become pairs (s^{p}, s^{q})

Merging Leaf Processes



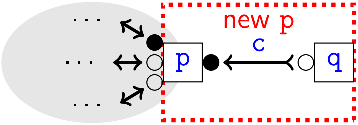
Idea:

Merge leaf process q into p by summarizing q's behavior

- Schedule q last : q moves only when p attempts to receive from c
- Communications between p and q become synchronizations $c\,! \cdot c\,?$
- States of p become pairs (s^{p}, s^{q})
- Rules $(s^{p}, c\,?, t^{p})$ of p become $((s^{p}, s^{q}), \mathtt{test}(\varphi), (t^{p}, t^{q}))$ where

$$\varphi \quad = \quad \exists u \exists z \cdot (u, c\,!, t^{q}) \in \Delta^{q} \ \wedge \ (s^{q}, z) \xrightarrow{*}_{q} (u, x)$$

Merging Leaf Processes



Idea:

Merge leaf process q into p by summarizing q's behavior

- Schedule q last : q moves only when p attempts to receive from c
- Communications between p and q become synchronizations $c\,!\cdot c\,?$
- States of p become pairs (s^{p}, s^{q})
- Rules $(s^{p}, c\,?, t^{p})$ of p become $((s^{p}, s^{q}), \mathtt{test}(\varphi), (t^{p}, t^{q}))$ where

$$\varphi \quad = \quad \exists u \exists z \cdot (u, c\,!, t^{q}) \in \Delta^{q} \ \wedge \ (s^{q}, z) \xrightarrow{*}_{q} (u, x)$$

- Use Presburger-definability of *post** for one-counter machines

Agenda

- ➪ Motivating Example: Sliding Window Protocol
- ➪ Systems of Communicating One-Counter Machines and their Topology Parametrized Reachability Problem
- ➪ Main Theorem: Proof of "Only If" Direction
- ➪ Main Theorem: Proof of "If" Direction
- ➪ Related/On-going/Future Work

Summary

- Formal model of Systems of Communicating One-Counter Machines

Summary

- Formal model of Systems of Communicating One-Counter Machines
- Their topology-parametrized reachability question

Summary

- ➪ Formal model of Systems of Communicating One-Counter Machines
- ➪ Their topology-parametrized reachability question
- ➪ Complete characterization of decidability in terms of topologies

Summary

- ➪ Formal model of Systems of Communicating One-Counter Machines
- ➪ Their topology-parametrized reachability question
- ➪ Complete characterization of decidability in terms of topologies
- ➪ Technical (side-)result:
  reachability relations of one-counter machines fall in "good" fragment of Presburger arithmetics

Related Works

- communicating finite state machines (CFSM) [Brand/Zafiropoulo '81, Pachl '82]

Related Works

- ▷ communicating finite state machines (CFSM) [Brand/Zafiropoulo '81, Pachl '82]
- ▷ characterize decidable topologies for mixed lossy & reliable channels [Chambart/Schnoebelen '08]

Related Works

- communicating finite state machines (CFSM)
  [Brand/Zafiropoulo '81, Pachl '82]
- characterize decidable topologies for mixed lossy & reliable channels
  [Chambart/Schnoebelen '08]
- CFSM with infinite message alphabets
  [Le Gall/Jeannet '07]

Related Works

- communicating finite state machines (CFSM)
  [Brand/Zafiropoulo '81, Pachl '82]
- characterize decidable topologies for mixed lossy & reliable channels
  [Chambart/Schnoebelen '08]
- CFSM with infinite message alphabets
  [Le Gall/Jeannet '07]
- (fifo-) communicating pushdown machines
  [LaTorre/Parlato/Madhusudan '08]

  and the influence of topologies on decidability
  [Heußner/Leroux/Muscholl/Sutre '10]

Related Works

- communicating finite state machines (CFSM)
  [Brand/Zafiropoulo '81, Pachl '82]
- characterize decidable topologies for mixed lossy & reliable channels
  [Chambart/Schnoebelen '08]
- CFSM with infinite message alphabets
  [Le Gall/Jeannet '07]
- (fifo-) communicating pushdown machines
  [LaTorre/Parlato/Madhusudan '08]

  and the influence of topologies on decidability
  [Heußner/Leroux/Muscholl/Sutre '10]
- decidable restrictions of multi-counter machines,
  model checking register machines,...

Decidability of Eager Reachability

Definition:

A full run ρ is eager if matching $(c\,!, c\,?)$ pairs are consecutive in ρ

If $\mathcal{T}$ is cycle-free,

- ➪ Full runs can be re-ordered into eager ones
- ➪ $\textsc{Rp-Sc1cm-Eager}(\mathcal{T})$ is decidable iff $\mathcal{T}$ is shunt-free

Proposition:

If $\mathcal{T}$ is strongly connected, then $\textsc{Rp-Sc1cm-Eager}(\mathcal{T})$ is decidable iff $\mathcal{T}$ contains at most two processes

Open: full characterization of decidable topologies (for eager reachability)

Perspectives

Complexity of RP-SC1CM for decidable topologies

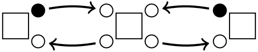
- ➪ At least PSPACE-hard

Perspectives

Complexity of Rp-Sc1cm for decidable topologies

➪ At least PSPACE-hard

Lossy channel Communicating One-Counter Machines

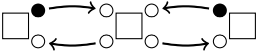
➪ Undecidable for  using acknowledgments

Perspectives

Complexity of Rp-Sc1cm for decidable topologies

➪ At least PSPACE-hard

Lossy channel Communicating One-Counter Machines

➪ Undecidable for  using acknowledgments

Extension from counters to stacks (i.e., send/receive stack)

Conjecture:

Rp-ScPDM($\mathcal{T}$) is decidable iff $\mathcal{T}$ is cycle-free and shunt-free