Foundations of the UML

Lecture 4: Message Passing Automata

Joost-Pieter Katoen

November 19, 2007

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p,q) \mid p,q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p,q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p,q) \mid p,q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p,q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p,q) \mid p,q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p,q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p, q) \mid p, q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p, q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p, q) \mid p, q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p, q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act_p^! := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act_p^? := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act_p^! \cup Act_p^?$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p, q) \mid p, q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p, q) \in Ch,\ a \in \mathbb{C}\}$

The architecture of a message-passing system

Definition

We fix the following parameters:

- $\mathcal{P}$ a finite set of at least two (sequential) processes
- $\mathbb{C}$ a finite set of message contents

Definition (communication actions, channels)

- $Act^!_p := \{p!q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$ (for $p \in \mathcal{P}$)
  "p sends message a to q"
- $Act^?_p := \{p?q(a) \mid q \in \mathcal{P} \setminus \{p\},\ a \in \mathbb{C}\}$
  "p receives message a from q"
- $Act_p := Act^!_p \cup Act^?_p$
- $Act := \bigcup_{p \in \mathcal{P}} Act_p$
- $Ch := \{(p, q) \mid p, q \in \mathcal{P},\ p \neq q\}$
- $Com := \{(p!q(a), q?p(a)) \mid (p, q) \in Ch,\ a \in \mathbb{C}\}$

Message-passing automata

Definition

A message-passing automaton (MPA) over $\mathcal{P}$ and $\mathbb{C}$ is a structure

$$\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$$

where

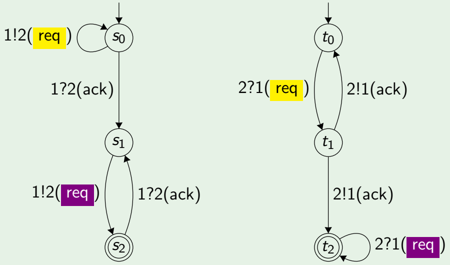
- $\mathbb{D}$ is a nonempty finite set of synchronization messages (or data)
- for each $p \in \mathcal{P}$
  - S_p is a nonempty finite set of local states (the S_p are disjoint)
  - $\Delta_p \subseteq S_p \times Act_p \times \mathbb{D} \times S_p$ is a set of local transitions
- $s_{init} \in S_{\mathcal{A}}$ is the global initial state
- $F \subseteq S_{\mathcal{A}}$ is the set of global final states

hereby: $S_{\mathcal{A}} := \prod_{p \in \mathcal{P}} S_p$ is the set of global states of $\mathcal{A}$

Note: We sometimes write $s \xrightarrow{\sigma, m}_p s'$ instead of $(s, \sigma, m, s') \in \Delta_p$.

Message-passing automata

Example



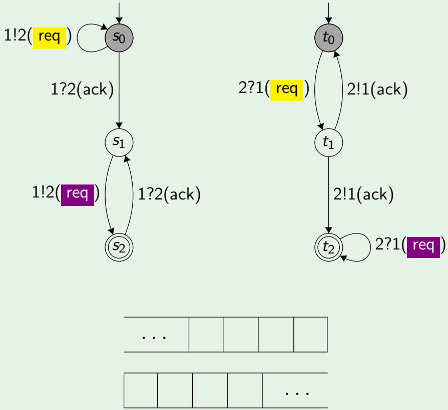
MPA $\mathcal{A}$ over $\{1, 2\}$ and $\{req, ack\}$

- $\mathbb{D} = \{$ yellow, purple, white $\}$
- $S_1 = \{s_0, s_1, s_2\}$
- $S_2 = \{t_0, t_1, t_2\}$
- Δ_1: $s_0 \xrightarrow{1!2(req)}_1 s_0 \ \ldots$

  Δ_2: $t_0 \xrightarrow{2?1(req)}_2 t_1 \ \ldots$
- $s_{init} = (s_0, t_0)$
- $F = \{(s_2, t_2)\}$

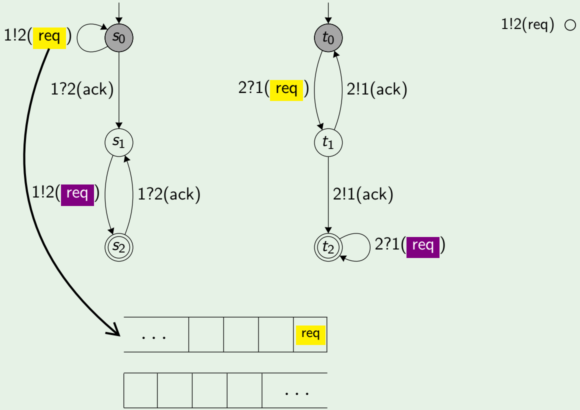
Message-passing automata

Example



Message-passing automata

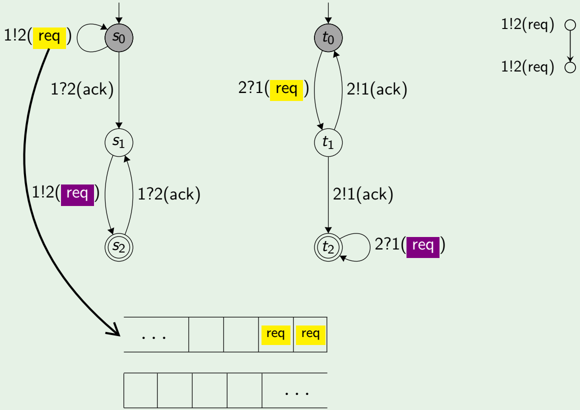
Example



1!2(req)

Message-passing automata

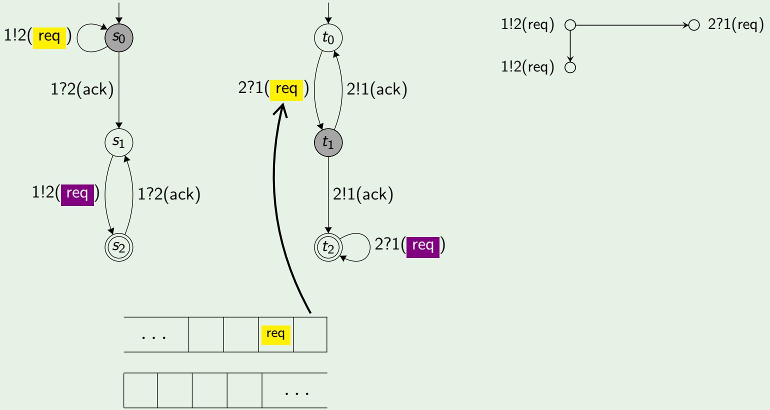
Example



1!2(req) 1!2(req)

Message-passing automata

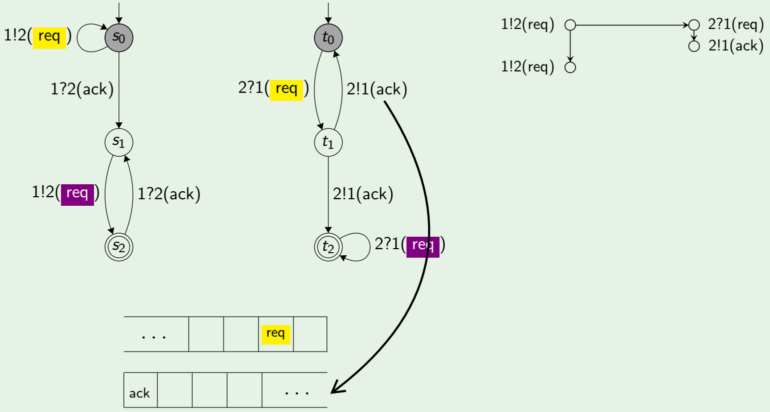
Example



1!2(req) 1!2(req) 2?1(req)

Message-passing automata

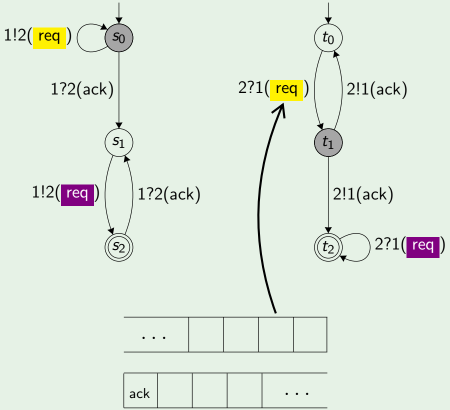
Example



1!2(req) 1!2(req) 2?1(req) 2!1(ack)

Message-passing automata

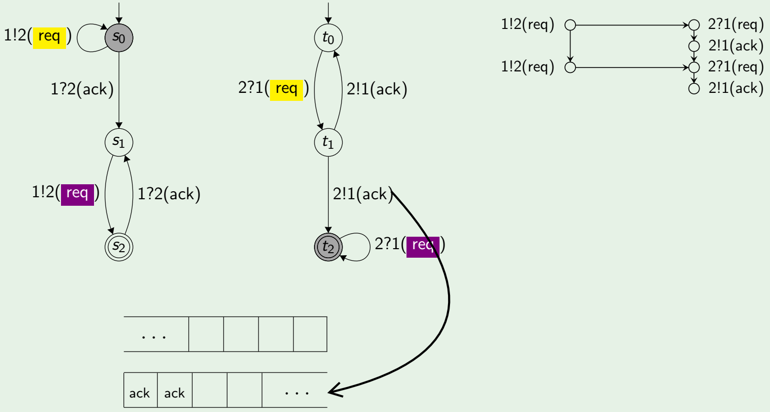
Example



1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req)

Message-passing automata

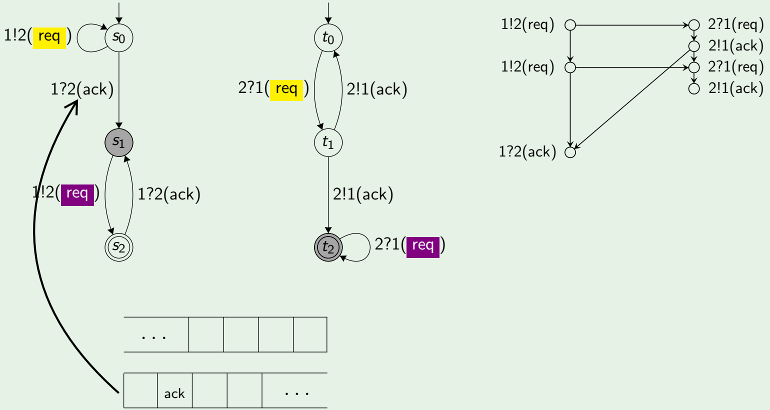
Example



1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack)

Message-passing automata

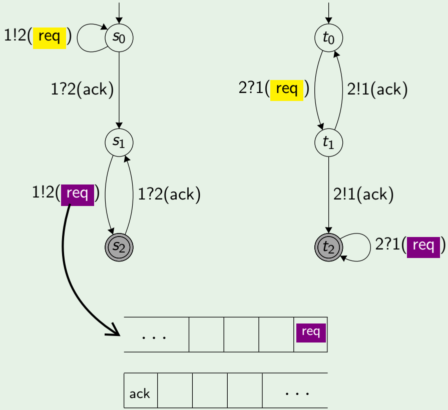
Example

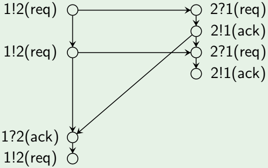


1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack)

Message-passing automata

Example

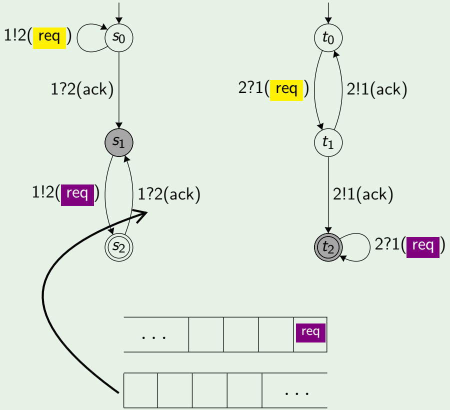


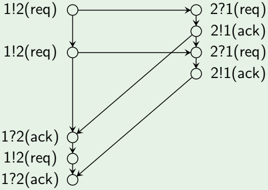


1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req)

Message-passing automata

Example

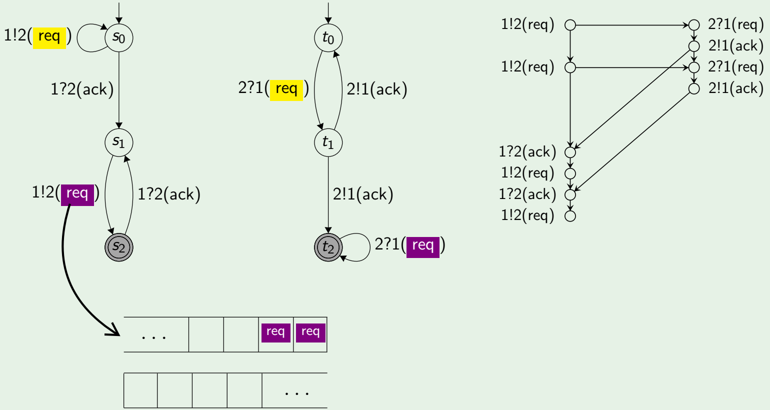




1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req) 1?2(ack)

Message-passing automata

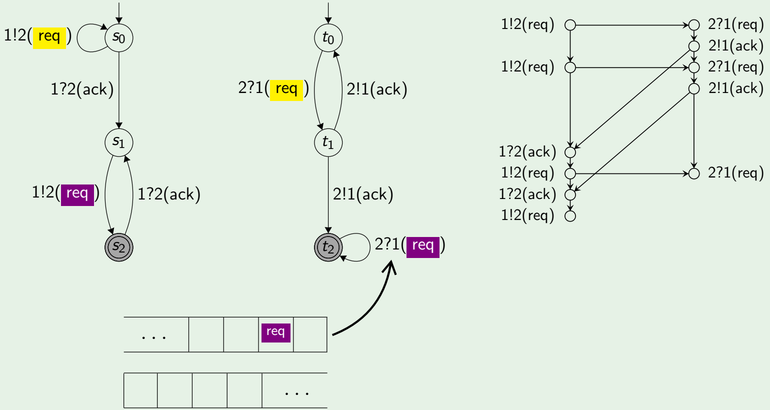
Example



1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req) 1?2(ack) 1!2(req)

Message-passing automata

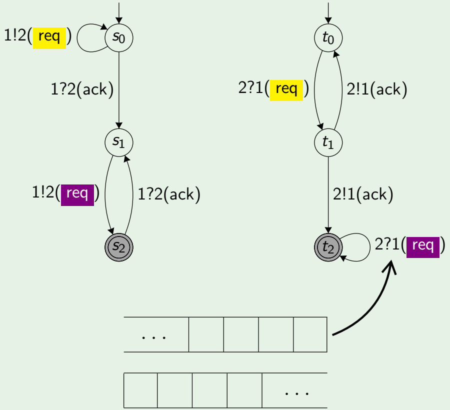
Example

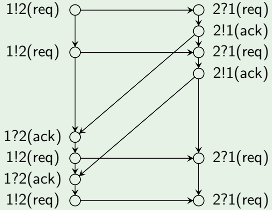


1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req) 1?2(ack) 1!2(req) 2?1(req)

Message-passing automata

Example

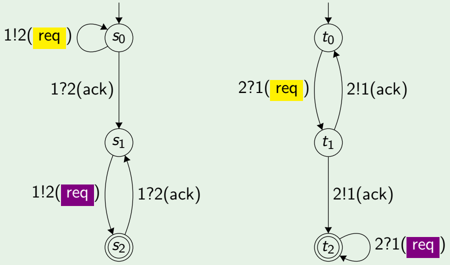


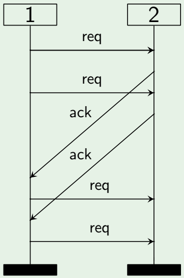


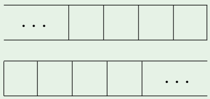
1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req) 1?2(ack) 1!2(req) 2?1(req) 2?1(req)

Message-passing automata

Example







1!2(req) 1!2(req) 2?1(req) 2!1(ack) 2?1(req) 2!1(ack) 1?2(ack) 1!2(req) 1?2(ack) 1!2(req) 2?1(req) 2?1(req)

Interpretation of MPAs

Let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over $\mathcal{P}$ and $\mathbb{C}$.

Definition

configurations of $\mathcal{A}$: $Conf_{\mathcal{A}} := S_{\mathcal{A}} \times \{\eta \mid \eta : Ch \to (\mathbb{C} \times \mathbb{D})^*\}$

Definition (global step)

$\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathbb{D} \times Conf_{\mathcal{A}}$ is defined as follows:

- sending a message: $((\overline{s}, \eta), p!q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p!q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta' = \eta[(p, q)/(a, m) \cdot \eta((p, q))]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$
- receipt of a message: $((\overline{s}, \eta), p?q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p?q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta = \eta'[(q, p)/\eta'((q, p)) \cdot (a, m)]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$

Interpretation of MPAs

Let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over $\mathcal{P}$ and $\mathbb{C}$.

Definition

configurations of $\mathcal{A}$: $Conf_{\mathcal{A}} := S_{\mathcal{A}} \times \{\eta \mid \eta : Ch \to (\mathbb{C} \times \mathbb{D})^*\}$

Definition (global step)

$\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathbb{D} \times Conf_{\mathcal{A}}$ is defined as follows:

- sending a message: $((\overline{s}, \eta), p!q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p!q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta' = \eta[(p,q)/(a,m) \cdot \eta((p,q))]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$
- receipt of a message: $((\overline{s}, \eta), p?q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p?q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta = \eta'[(q,p)/\eta'((q,p)) \cdot (a,m)]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$

Interpretation of MPAs

Let $\mathcal{A} = (((S_p, \Delta_p))_{p\in\mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over $\mathcal{P}$ and $\mathbb{C}$.

Definition

configurations of $\mathcal{A}$: $Conf_{\mathcal{A}} := S_{\mathcal{A}} \times \{\eta \mid \eta : Ch \to (\mathbb{C} \times \mathbb{D})^*\}$

Definition (global step)

$\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathbb{D} \times Conf_{\mathcal{A}}$ is defined as follows:

- sending a message: $((\overline{s}, \eta), p!q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p!q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta' = \eta[(p,q)/(a,m) \cdot \eta((p,q))]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$
- receipt of a message: $((\overline{s}, \eta), p?q(a), m, (\overline{s}', \eta')) \in \Longrightarrow_{\mathcal{A}}$ if
  - $(\overline{s}[p], p?q(a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta = \eta'[(q,p)/\eta'((q,p)) \cdot (a,m)]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$

Linearizations of an MPA

Let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over over $\mathcal{P}$ and $\mathbb{C}$.

Definition

A run of $\mathcal{A}$ on $\sigma_1 \ldots \sigma_n \in Act^*$ is a sequence $\rho = \gamma_0\, m_1\, \gamma_1 \ldots \gamma_{n-1}\, m_n\, \gamma_n$ such that

- $\gamma_0 = (s_{init}, \eta_\varepsilon)$ with η_ε mapping any channel to ε
- $\gamma_{i-1} \xrightarrow{\sigma_i, m_i}_{\mathcal{A}} \gamma_i$ for any $i \in \{1, \ldots, n\}$

Run ρ is accepting if $\gamma_n \in F \times \{\eta_\varepsilon\}$.

Definition

The set of linearizations of $\mathcal{A}$:

$Lin(\mathcal{A}) := \{w \in Act^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$

Linearizations of an MPA

Let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over over $\mathcal{P}$ and $\mathbb{C}$.

Definition

A run of $\mathcal{A}$ on $\sigma_1 \ldots \sigma_n \in Act^*$ is a sequence $\rho = \gamma_0\, m_1\, \gamma_1 \ldots \gamma_{n-1}\, m_n\, \gamma_n$ such that

- $\gamma_0 = (s_{init}, \eta_\varepsilon)$ with η_ε mapping any channel to ε
- $\gamma_{i-1} \xrightarrow{\sigma_i, m_i}_{\mathcal{A}} \gamma_i$ for any $i \in \{1, \ldots, n\}$

Run ρ is accepting if $\gamma_n \in F \times \{\eta_\varepsilon\}$.

Definition

The set of linearizations of $\mathcal{A}$:

$Lin(\mathcal{A}) := \{w \in Act^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$

Linearizations of an MPA

Let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be an MPA over over $\mathcal{P}$ and $\mathbb{C}$.

Definition

A run of $\mathcal{A}$ on $\sigma_1 \ldots \sigma_n \in Act^*$ is a sequence $\rho = \gamma_0\, m_1\, \gamma_1 \ldots \gamma_{n-1}\, m_n\, \gamma_n$ such that

- $\gamma_0 = (s_{init}, \eta_\varepsilon)$ with η_ε mapping any channel to ε
- $\gamma_{i-1} \xrightarrow{\sigma_i, m_i}_{\mathcal{A}} \gamma_i$ for any $i \in \{1, \ldots, n\}$

Run ρ is accepting if $\gamma_n \in F \times \{\eta_\varepsilon\}$.

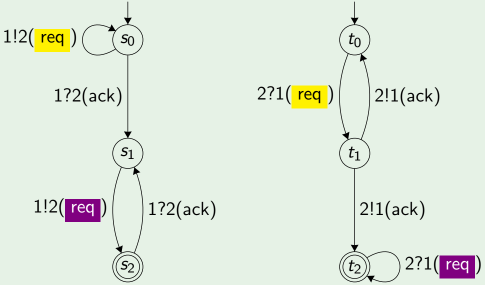
Definition

The set of linearizations of $\mathcal{A}$:

$Lin(\mathcal{A}) := \{w \in Act^* \mid \text{there is an accepting run of } \mathcal{A} \text{ on } w\}$

Linearizations of an example MPA

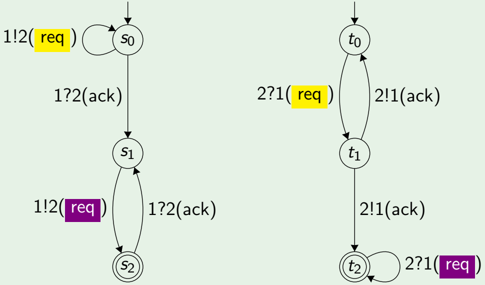
Example



MPA $\mathcal{A}$ over $\{1, 2\}$ and $\{req, ack\}$

Linearizations of an example MPA

Example



MPA $\mathcal{A}$ over $\{1,2\}$ and $\{\text{req}, \text{ack}\}$

$Lin(\mathcal{A}) = \Big\{ w \in Act^* \mid$ there is $n \geq 1$ such that:

$w \upharpoonright 1 = (1!2(\text{req}))^n \ (1?2(\text{ack}) \ 1!2(\text{req}))^n$

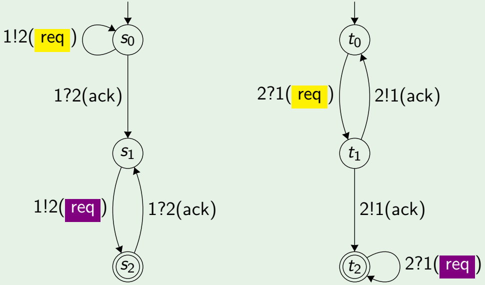
$w \upharpoonright 2 = (2?1(\text{req}) \ 2!1(\text{ack}))^n \ (2?1(\text{req}))^n$

for any $u \in Pref(w)$ and $(p,q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \geq 0 \Big\}$$

Linearizations of an example MPA

Example



MPA $\mathcal{A}$ over $\{1,2\}$ and $\{\text{req},\text{ack}\}$

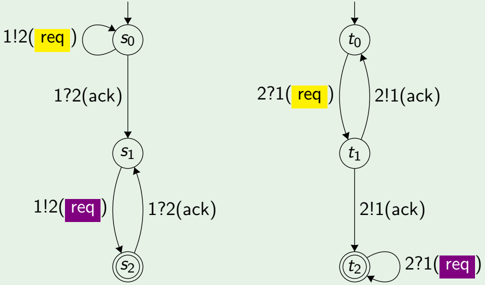
- 1!2(req) and 2!1(ack) are always independent.
- 1!2(req) and 1?2(ack) are always dependent.
- 1!2(req) and 2?1(req) are sometimes independent.

⇝ non-regular (word) languages

⇝ actually more complicated than framework of traces

Linearizations and MSCs of an example MPA

Example



MPA $\mathcal{A}$ over $\{1,2\}$ and $\{req, ack\}$

$$Lin(\mathcal{A}) = \Big\{ w \in Act^* \mid \text{there is } n \geq 1 \text{ such that:}$$

$$w \restriction 1 = (1!2(req))^n \; (1?2(ack)\; 1!2(req))^n$$

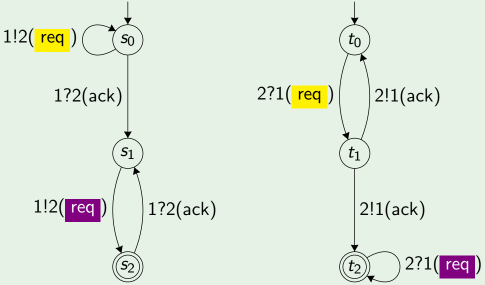
$$w \restriction 2 = (2?1(req)\; 2!1(ack))^n \; (2?1(req))^n$$

for any $u \in Pref(w)$ and $(p,q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \geq 0 \Big\}$$

Linearizations and MSCs of an example MPA

Example



MPA $\mathcal{A}$ over $\{1, 2\}$ and $\{\text{req}, \text{ack}\}$

$$L(\mathcal{A}) = \Big\{ \mathcal{M} \in \mathbb{M} \mid \text{there is } n \geq 1 \text{ such that:}$$

$$\mathcal{M} \restriction 1 = (1!2(\text{req}))^n \; (1?2(\text{ack}) \; 1!2(\text{req}))^n$$

$$\mathcal{M} \restriction 2 = (2?1(\text{req}) \; 2!1(\text{ack}))^n \; (2?1(\text{req}))^n \Big\}$$

Elementary questions are undecidable for MPA ...

Proposition (Brand & Zafiropulo 1983)

The following problem is undecidable (even if $\mathbb{C}$ is a singleton):

INPUT:	MPA $\mathcal{A}$ over $\mathcal{P}$ and $\mathbb{C}$
QUESTION:	Is $L(\mathcal{A})$ empty?

Proof (sketch)

Reduction from halting problem for Turing machine $TM = (Q, \Sigma, \Delta, \Box, q_0, q_f)$ to emptiness for MPA with two processes. Build MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ over $\{1, 2\}$ and some singleton set such that $L(\mathcal{A}) \neq \emptyset$ iff TM can reach q_f.

- Process 1 sends current configurations to process 2.
- Process 2 chooses successor configurations and sends them back to 1.
- $\mathbb{D} = \Big((\Sigma \cup \{\Box\}) \times (Q \cup \{\_\})\Big) \cup \{\#\}$

Elementary questions are undecidable for MPA ...

Proposition (Brand & Zafiropulo 1983)

The following problem is undecidable (even if $\mathbb{C}$ is a singleton):

INPUT:	MPA $\mathcal{A}$ over $\mathcal{P}$ and $\mathbb{C}$
QUESTION:	Is $L(\mathcal{A})$ empty?

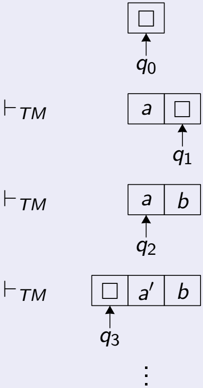
Proof (sketch)

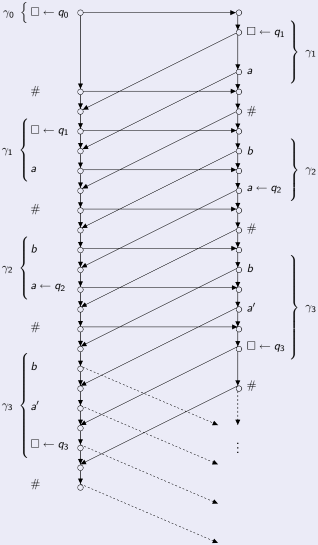
Reduction from halting problem for Turing machine $TM = (Q, \Sigma, \Delta, \Box, q_0, q_f)$ to emptiness for MPA with two processes. Build MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ over $\{1, 2\}$ and some singleton set such that $L(\mathcal{A}) \neq \emptyset$ iff TM can reach q_f.

- Process 1 sends current configurations to process 2.
- Process 2 chooses successor configurations and sends them back to 1.
- $\mathbb{D} = \Big((\Sigma \cup \{\Box\}) \times (Q \cup \{\_\})\Big) \cup \{\#\}$

An MPA simulating a Turing machine

Proof (contd.)





An MPA simulating a Turing machine

Proof (contd.)

- Left or standstill transition: Process 2 may just wait for a symbol containing a state of the Turing machine and to alter it correspondingly. In the example, the left-moving transition (q_2, a, a', L, q_3) is applied so that process 2
  - sends b unchanged back to process 1
  - detects (receives) $a \leftarrow q_2$
  - sends a' to process 1 entering a state indicating that the symbol to be sent next has to be equipped with q_3
  - receives $\#$ so that the symbol $\square \leftarrow q_3$ has to be inserted before returning $\#$
- Right transition: Process 2 has to guess what the position right before the head is. For example, provided process 2 decided in favor of (q_2, a, a', R, q_3) while reading the b, it would have to
  - send $b \leftarrow q_3$ instead of just b, entering some state $t(a \leftarrow q_2)$
  - receive $a \leftarrow q_2$ (no other symbol can be received in state $t(a \leftarrow q_2)$)
  - send a' back to process 1

An MPA simulating a Turing machine

Proof (contd.)

- Left or standstill transition: Process 2 may just wait for a symbol containing a state of the Turing machine and to alter it correspondingly. In the example, the left-moving transition (q_2, a, a', L, q_3) is applied so that process 2
  - sends b unchanged back to process 1
  - detects (receives) $a \leftarrow q_2$
  - sends a' to process 1 entering a state indicating that the symbol to be sent next has to be equipped with q_3
  - receives $\#$ so that the symbol $\square \leftarrow q_3$ has to be inserted before returning $\#$
- Right transition: Process 2 has to guess what the position right before the head is. For example, provided process 2 decided in favor of (q_2, a, a', R, q_3) while reading the b, it would have to
  - send $b \leftarrow q_3$ instead of just b, entering some state $t(a \leftarrow q_2)$
  - receive $a \leftarrow q_2$ (no other symbol can be received in state $t(a \leftarrow q_2)$)
  - send a' back to process 1

An MPA simulating a Turing machine

Proof (contd.)

- Introduce local final states s_f and t_f, one for process 1 and one for process 2, respectively (i.e., $F = \{(s_f, t_f)\}$ and $\mathcal{A}$ is locally accepting).
- At any time, process 1 may switch into s_f, in which arbitrary and arbitrarily many messages can be received to empty channel $(2, 1)$.
- Process 2 is allowed to move into t_f and to empty the channel $(1, 2)$ as soon as it receives a letter $c \leftarrow q_f$ for some c.
- As process 2 modifies a configuration of TM locally, finitely many states are sufficient in $\mathcal{A}$. □

An MPA simulating a Turing machine

Proof (contd.)

- Introduce local final states s_f and t_f, one for process 1 and one for process 2, respectively (i.e., $F = \{(s_f, t_f)\}$ and $\mathcal{A}$ is locally accepting).
- At any time, process 1 may switch into s_f, in which arbitrary and arbitrarily many messages can be received to empty channel $(2, 1)$.
- Process 2 is allowed to move into t_f and to empty the channel $(1, 2)$ as soon as it receives a letter $c \leftarrow q_f$ for some c.
- As process 2 modifies a configuration of TM locally, finitely many states are sufficient in $\mathcal{A}$. □

An MPA simulating a Turing machine

Proof (contd.)

- Introduce local final states s_f and t_f, one for process 1 and one for process 2, respectively (i.e., $F = \{(s_f, t_f)\}$ and $\mathcal{A}$ is locally accepting).
- At any time, process 1 may switch into s_f, in which arbitrary and arbitrarily many messages can be received to empty channel $(2, 1)$.
- Process 2 is allowed to move into t_f and to empty the channel $(1, 2)$ as soon as it receives a letter $c \leftarrow q_f$ for some c.
- As process 2 modifies a configuration of TM locally, finitely many states are sufficient in $\mathcal{A}$. □

An MPA simulating a Turing machine

Proof (contd.)

- Introduce local final states s_f and t_f, one for process 1 and one for process 2, respectively (i.e., $F = \{(s_f, t_f)\}$ and $\mathcal{A}$ is locally accepting).
- At any time, process 1 may switch into s_f, in which arbitrary and arbitrarily many messages can be received to empty channel $(2, 1)$.
- Process 2 is allowed to move into t_f and to empty the channel $(1, 2)$ as soon as it receives a letter $c \leftarrow q_f$ for some c.
- As process 2 modifies a configuration of TM locally, finitely many states are sufficient in $\mathcal{A}$. □

Towards subclasses of MPA: Boundedness

Definition (B-bounded words)

Let $B \geq 1$. A word $w \in Act^*$ is called B-bounded if, for any $u \in Pref(w)$ and any channel $(p, q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \leq B$$

Example

$1!2(a)\ 1!2(b)\ 2?1(a)\ 2?1(b)$ is 2-bounded but not 1-bounded.

Definition (bounded MSCs)

Let $B \geq 1$. An MSC $\mathcal{M} \in \mathbb{M}$ is called

- universally B-bounded ($\forall B$-bounded) if $Lin(\mathcal{M}) = Lin^B(\mathcal{M})$ where $Lin^B(\mathcal{M}) := \{w \in Lin(\mathcal{M}) \mid w \text{ is } B\text{-bounded}\}$.
- existentially B-bounded ($\exists B$-bounded) if $Lin(\mathcal{M}) \cap Lin^B(\mathcal{M}) \neq \emptyset$.

Towards subclasses of MPA: Boundedness

Definition (B-bounded words)

Let $B \geq 1$. A word $w \in Act^*$ is called B-bounded if, for any $u \in Pref(w)$ and any channel $(p, q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \leq B$$

Example

$1!2(a)\ 1!2(b)\ 2?1(a)\ 2?1(b)$ is 2-bounded but not 1-bounded.

Definition (bounded MSCs)

Let $B \geq 1$. An MSC $\mathcal{M} \in \mathbb{M}$ is called

- universally B-bounded ($\forall B$-bounded) if $Lin(\mathcal{M}) = Lin^B(\mathcal{M})$ where $Lin^B(\mathcal{M}) := \{w \in Lin(\mathcal{M}) \mid w \text{ is } B\text{-bounded}\}$,
- existentially B-bounded ($\exists B$-bounded) if $Lin(\mathcal{M}) \cap Lin^B(\mathcal{M}) \neq \emptyset$.

Towards subclasses of MPA: Boundedness

Definition (B-bounded words)

Let $B \geq 1$. A word $w \in Act^*$ is called B-bounded if, for any $u \in Pref(w)$ and any channel $(p, q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \leq B$$

Example

$1!2(a)\ 1!2(b)\ 2?1(a)\ 2?1(b)$ is 2-bounded but not 1-bounded.

Definition (bounded MSCs)

Let $B \geq 1$. An MSC $\mathcal{M} \in \mathbb{M}$ is called

- universally B-bounded ($\forall B$-bounded) if $Lin(\mathcal{M}) = Lin^B(\mathcal{M})$ where $Lin^B(\mathcal{M}) := \{w \in Lin(\mathcal{M}) \mid w \text{ is } B\text{-bounded}\}$.
- existentially B-bounded ($\exists B$-bounded) if $Lin(\mathcal{M}) \cap Lin^B(\mathcal{M}) \neq \emptyset$.

Towards subclasses of MPA: Boundedness

Definition (B-bounded words)

Let $B \geq 1$. A word $w \in Act^*$ is called B-bounded if, for any $u \in Pref(w)$ and any channel $(p, q) \in Ch$:

$$\sum_{a \in \mathbb{C}} |u|_{p!q(a)} - \sum_{a \in \mathbb{C}} |u|_{q?p(a)} \leq B$$

Example

$1!2(a)\ 1!2(b)\ 2?1(a)\ 2?1(b)$ is 2-bounded but not 1-bounded.

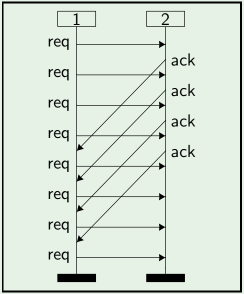
Definition (bounded MSCs)

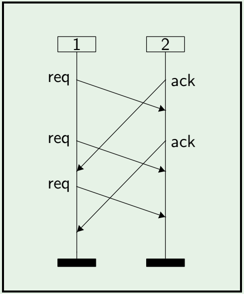
Let $B \geq 1$. An MSC $\mathcal{M} \in \mathbb{M}$ is called

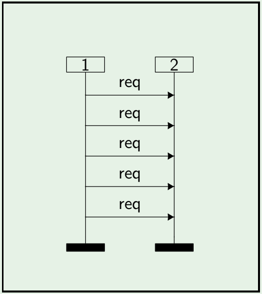
- universally B-bounded ($\forall B$-bounded) if $Lin(\mathcal{M}) = Lin^B(\mathcal{M})$ where $Lin^B(\mathcal{M}) := \{w \in Lin(\mathcal{M}) \mid w \text{ is } B\text{-bounded}\}$.
- existentially B-bounded ($\exists B$-bounded) if $Lin(\mathcal{M}) \cap Lin^B(\mathcal{M}) \neq \emptyset$.

Bounded MSCs

Example

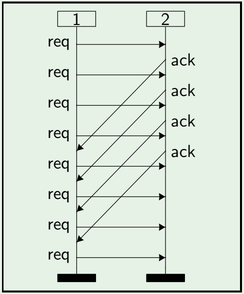


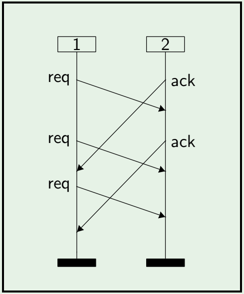


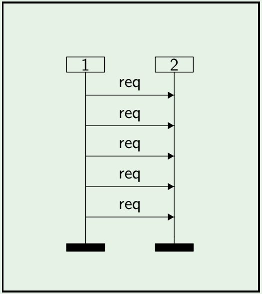


Bounded MSCs

Example







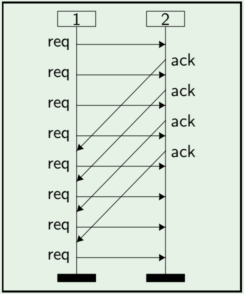
∀4-bounded

∃2-bounded

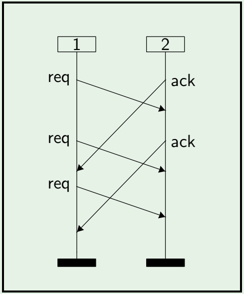
not ∃1-bounded

Bounded MSCs

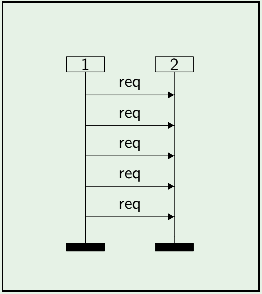
Example



∀4-bounded
∃2-bounded
not ∃1-bounded

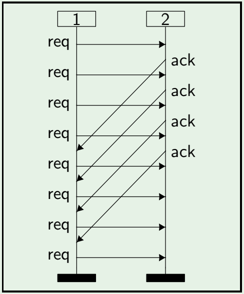


∀3-bounded
∃1-bounded

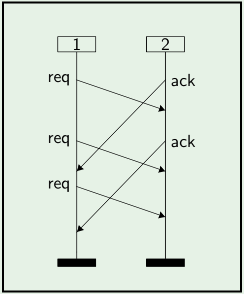


Bounded MSCs

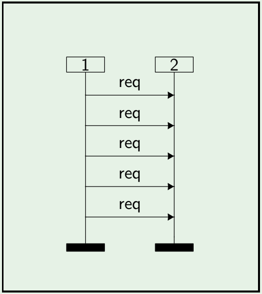
Example



∀4-bounded
∃2-bounded
not ∃1-bounded



∀3-bounded
∃1-bounded



∀5-bounded
∃1-bounded

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

Definition

An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable

hereby: $\mathbb{M}_{\forall B}$ is the set of $\forall B$-bounded MSCs
$\mathbb{M}_{\exists B}$ is the set of $\exists B$-bounded MSCs

A zoo of MPA

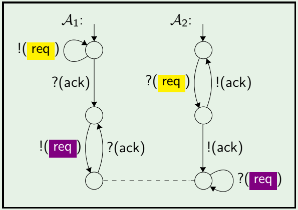
Definition

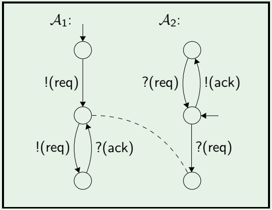
An MPA $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ over $\mathcal{P}$ and $\mathbb{C}$ is called

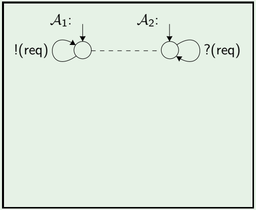
- $\forall B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\forall B}$.
- strongly B-bounded if, for any $u \in Act^*$:
  if there is a run of $\mathcal{A}$ on u, then u is B-bounded.
- $\exists B$-bounded if $L(\mathcal{A}) \subseteq \mathbb{M}_{\exists B}$.
- a product MPA if $|\mathbb{D}| = 1$.
- locally accepting if $F = \prod_{p \in \mathcal{P}} F_p$ for some sets $F_p \subseteq S_p$, $p \in \mathcal{P}$.
- deadlock-free if from every reachable configuration a final configuration is reachable
- deterministic if the following holds:
  - if $s \xrightarrow{p!q(a),m_1}_p s_1$ and $s \xrightarrow{p!q(a),m_2}_p s_2$, then $s_1 = s_2$ and $m_1 = m_2$
  - if $s \xrightarrow{p?q(a),m}_p s_1$ and $s \xrightarrow{p?q(a),m}_p s_2$, then $s_1 = s_2$

A zoo of MPA

Example

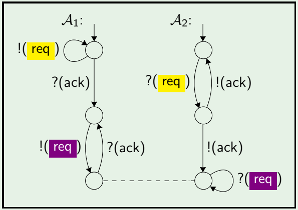


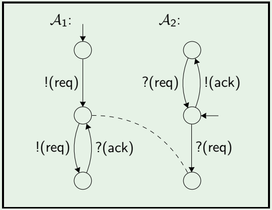


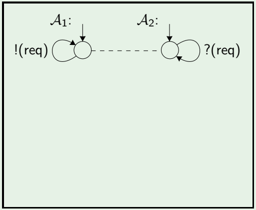


A zoo of MPA

Example







not $\exists B$-bounded f.a. B

not a product MPA

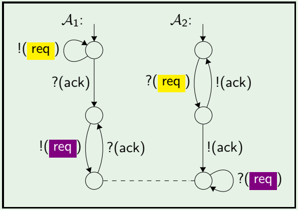
locally accepting

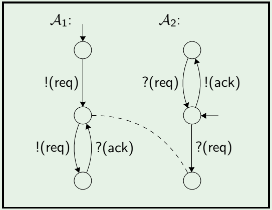
not safe

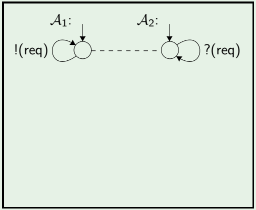
not deterministic

A zoo of MPA

Example





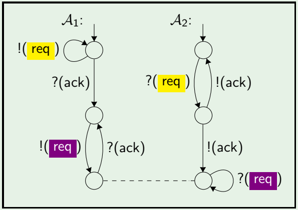


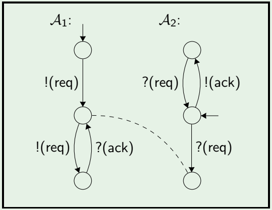
not $\exists B$-bounded f.a. B
not a product MPA
locally accepting
not safe
not deterministic

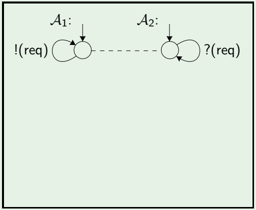
strongly $\forall 3$-bounded
product MPA
locally accepting
safe
deterministic

A zoo of MPA

Example







not $\exists B$-bounded f.a. B	strongly $\forall 3$-bounded	not $\forall B$-bound.f.a.B
not a product MPA	product MPA	$\exists 1$-bounded
locally accepting	locally accepting	product MPA
not safe	safe	locally accepting
not deterministic	deterministic	safe
		deterministic

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m, n) \in \mathbb{M}$ over $(\{1, 2\}, \{\mathrm{req}, \mathrm{ack}\})$ be given by:

- $\mathcal{M} \upharpoonright 1 = (1!2(\mathrm{req}))^m \; (1?2(\mathrm{ack}) \; 1!2(\mathrm{req}))^n$
- $\mathcal{M} \upharpoonright 2 = (2?1(\mathrm{req}) \; 2!1(\mathrm{ack}))^n \; (2?1(\mathrm{req}))^m$

There is no product MPA over $\{1, 2\}$ and $\{\mathrm{req}, \mathrm{ack}\}$ whose language is $L = \{\mathcal{M}(n, n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n, n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n, n) \upharpoonright 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n, n) \upharpoonright 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m,n) \in \mathbb{M}$ over $(\{1,2\}, \{\mathsf{req}, \mathsf{ack}\})$ be given by:

- $\mathcal{M} \upharpoonright 1 = (1!2(\mathsf{req}))^m \ (1?2(\mathsf{ack}) \ 1!2(\mathsf{req}))^n$
- $\mathcal{M} \upharpoonright 2 = (2?1(\mathsf{req}) \ 2!1(\mathsf{ack}))^n \ (2?1(\mathsf{req}))^m$

There is no product MPA over $\{1,2\}$ and $\{\mathsf{req}, \mathsf{ack}\}$ whose language is $L = \{\mathcal{M}(n,n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n,n) \upharpoonright 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n,n) \upharpoonright 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m,n) \in \mathbb{M}$ over $(\{1,2\}, \{\mathsf{req}, \mathsf{ack}\})$ be given by:

- $\mathcal{M} \upharpoonright 1 = (1!2(\mathsf{req}))^m \; (1?2(\mathsf{ack}) \; 1!2(\mathsf{req}))^n$
- $\mathcal{M} \upharpoonright 2 = (2?1(\mathsf{req}) \; 2!1(\mathsf{ack}))^n \; (2?1(\mathsf{req}))^m$

There is no product MPA over $\{1,2\}$ and $\{\mathsf{req}, \mathsf{ack}\}$ whose language is $L = \{\mathcal{M}(n,n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n,n) \upharpoonright 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n,n) \upharpoonright 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m,n) \in \mathbb{M}$ over $(\{1,2\}, \{\mathsf{req}, \mathsf{ack}\})$ be given by:

- $\mathcal{M}\restriction 1 = (1!2(\mathsf{req}))^m \ (1?2(\mathsf{ack})\ 1!2(\mathsf{req}))^n$
- $\mathcal{M}\restriction 2 = (2?1(\mathsf{req})\ 2!1(\mathsf{ack}))^n \ (2?1(\mathsf{req}))^m$

There is no product MPA over $\{1,2\}$ and $\{\mathsf{req}, \mathsf{ack}\}$ whose language is $L = \{\mathcal{M}(n,n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n,n)\restriction 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n,n)\restriction 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m,n) \in \mathbb{M}$ over $(\{1,2\}, \{\text{req}, \text{ack}\})$ be given by:

- $\mathcal{M} \restriction 1 = (1!2(\text{req}))^m \; (1?2(\text{ack}) \; 1!2(\text{req}))^n$
- $\mathcal{M} \restriction 2 = (2?1(\text{req}) \; 2!1(\text{ack}))^n \; (2?1(\text{req}))^m$

There is no product MPA over $\{1,2\}$ and $\{\text{req}, \text{ack}\}$ whose language is $L = \{\mathcal{M}(n,n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n,n) \restriction 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n,n) \restriction 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m,n) \in \mathbb{M}$ over $(\{1,2\}, \{\mathsf{req}, \mathsf{ack}\})$ be given by:

- $\mathcal{M} \upharpoonright 1 = (1!2(\mathsf{req}))^m \; (1?2(\mathsf{ack}) \; 1!2(\mathsf{req}))^n$
- $\mathcal{M} \upharpoonright 2 = (2?1(\mathsf{req}) \; 2!1(\mathsf{ack}))^n \; (2?1(\mathsf{req}))^m$

There is no product MPA over $\{1,2\}$ and $\{\mathsf{req}, \mathsf{ack}\}$ whose language is $L = \{\mathcal{M}(n,n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n,n) \upharpoonright 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n,n) \upharpoonright 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □

MPA vs. product MPA

Lemma

Product MPA are less expressive than MPA.

Proof.

For $m, n \geq 1$, let $\mathcal{M}(m, n) \in \mathbb{M}$ over $(\{1, 2\}, \{\mathsf{req}, \mathsf{ack}\})$ be given by:

- $\mathcal{M} \upharpoonright 1 = (1!2(\mathsf{req}))^m \; (1?2(\mathsf{ack}) \; 1!2(\mathsf{req}))^n$
- $\mathcal{M} \upharpoonright 2 = (2?1(\mathsf{req}) \; 2!1(\mathsf{ack}))^n \; (2?1(\mathsf{req}))^m$

There is no product MPA over $\{1, 2\}$ and $\{\mathsf{req}, \mathsf{ack}\}$ whose language is $L = \{\mathcal{M}(n, n) \mid n \geq 1\}$. Suppose there is a product MPA $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n \geq 1$, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n, n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i \geq 1$ to read the first n letters of $\mathcal{M}(n, n) \upharpoonright 1$
- $\mathcal{A}_2$ visits a cycle of length $j \geq 1$ to read the last n letters of $\mathcal{M}(n, n) \upharpoonright 2$

But then, there is an accepting run of $\mathcal{A}$ on $\mathcal{M}(n + (i \cdot j), n) \notin L$. □