Theoretical Foundations of the UML

Lecture 9: Bounded MSC and CFMs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

moves.rwth-aachen.de/teaching/ss-16/theoretical-foundations-of-the-uml/

29. Mai 2016

Outline

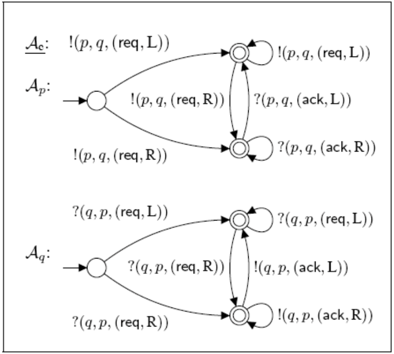
Overview

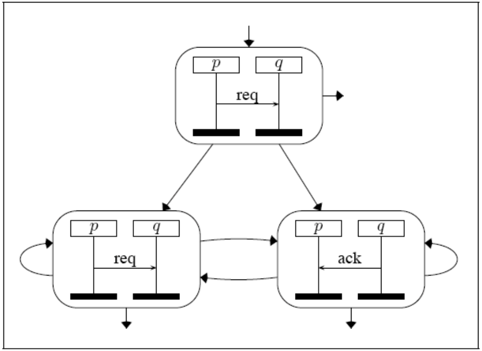
1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs

Communicating finite-state machines

- A communicating finite-state machine (CFM) is a collection of finite-state machines, one for each process
- Communication between these machines takes place via (a priori) unbounded reliable FIFO channels
- The underlying system architecture is parametrised by the set $\mathcal{P}$ of processes and the set $\mathcal{C}$ of messages
- Action $!(p, q, m)$ puts message m at the end of the channel (p, q)
- Action $?(q, p, m)$ is enabled only if m is at head of buffer, and its execution by process q removes m from the channel (p, q)
- Synchronisation messages are used to avoid deadlocks

Example communicating finite-state machine





This CFM accepts if $\mathcal{A}_p$ and $\mathcal{A}_q$ are in some local state, and (as usual) all channels are empty

Formal definition

Definition (What is a CFM?)

A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple

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

where

- for each $p \in \mathcal{P}$:

Formal definition

Definition (What is a CFM?)

A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple

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

where

- for each $p \in \mathcal{P}$:
  - S_p is a non-empty 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

In sequel, let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be a CFM over $\mathcal{P}$ and $\mathcal{C}$.

Formal definition

Definition (What is a CFM?)

A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple

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

where

- for each $p \in \mathcal{P}$:
  - S_p is a non-empty 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
- $\mathbb{D}$ is a nonempty finite set of synchronization messages (or data)

In sequel, let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be a CFM over $\mathcal{P}$ and $\mathcal{C}$.

Formal definition

Definition (What is a CFM?)

A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple

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

where

- for each $p \in \mathcal{P}$:
  - S_p is a non-empty 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
- $\mathbb{D}$ is a nonempty finite set of synchronization messages (or data)
- $s_{init} \in S_{\mathcal{A}}$ is the global initial state
  - where $S_{\mathcal{A}} := \prod_{p \in \mathcal{P}} S_p$ is the set of global states of $\mathcal{A}$

In sequel, let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be a CFM over $\mathcal{P}$ and $\mathcal{C}$.

Formal definition

Definition (What is a CFM?)

A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple

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

where

- for each $p \in \mathcal{P}$:
  - S_p is a non-empty 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
- $\mathbb{D}$ is a nonempty finite set of synchronization messages (or data)
- $s_{init} \in S_{\mathcal{A}}$ is the global initial state
  - where $S_{\mathcal{A}} := \prod_{p \in \mathcal{P}} S_p$ is the set of global states of $\mathcal{A}$
- $F \subseteq S_{\mathcal{A}}$ is the set of global final states

In sequel, let $\mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathbb{D}, s_{init}, F)$ be a CFM over $\mathcal{P}$ and $\mathcal{C}$.

Formal semantics of CFMs

Definition (Configuration)

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

Formal semantics of CFMs

Definition (Configuration)

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

Definition (Transitions between configurations)

$\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathbb{D} \times Conf_{\mathcal{A}}$ 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 $r \in \mathcal{P} \setminus \{p\}$

Formal semantics of CFMs

Definition (Configuration)

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

Definition (Transitions between configurations)

$\Longrightarrow_{\mathcal{A}} \subseteq Conf_{\mathcal{A}} \times Act \times \mathbb{D} \times Conf_{\mathcal{A}}$ 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
  - $(\overline{s}[p], ?(p, q, a), m, \overline{s}'[p]) \in \Delta_p$
  - $\eta((q, p)) = w \cdot (a, m) \neq \epsilon$ and $\eta' = \eta[(q, p) := w]$
  - $\overline{s}[r] = \overline{s}'[r]$ for all $r \in \mathcal{P} \setminus \{p\}$

Linearizations of a CFM

Definition ((Accepting) Runs)

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 $i \in \{1, \ldots, n\}$

Linearizations of a CFM

Definition ((Accepting) Runs)

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 $\gamma_n \in F \times \{\eta_\varepsilon\}$.

Linearizations of a CFM

Definition ((Accepting) Runs)

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 (Linearizations)

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

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

Overview

1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs

Well-formedness (reminder)

Let $Ch := \{(p,q) \mid p \neq q\,,\, p,q \in \mathcal{P}\}$ be a set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.: $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

Well-formedness (reminder)

Let $Ch := \{(p,q) \mid p \neq q\,,\ p,q \in \mathcal{P}\}$ be a set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

   where $|u|_a$ denotes the number of occurrences of action a in u

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leqslant i < j \leqslant n$, $(p,q) \in Ch$, and $a_i = !(p,q,m_1)$, $a_j = ?(q,p,m_2)$:

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)} \quad \text{implies} \quad m_1 = m_2$$

Well-formedness (reminder)

Let $Ch := \{(p,q) \mid p \neq q\,,\ p,q \in \mathcal{P}\}$ be a set of channels over $\mathcal{P}$.

We call $w = a_1 \ldots a_n \in Act^*$ proper if

1. every receive in w is preceded by a corresponding send, i.e.:
   $\forall (p,q) \in Ch$ and prefix u of w, we have:

$$\underbrace{\sum_{m \in \mathcal{C}} |u|_{!(p,q,m)}}_{\#\text{ sends from } p \text{ to } q} \quad \geqslant \quad \underbrace{\sum_{m \in \mathcal{C}} |u|_{?(q,p,m)}}_{\#\text{ receipts by } q \text{ from } p}$$

   where $|u|_a$ denotes the number of occurrences of action a in u

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leqslant i < j \leqslant n$, $(p,q) \in Ch$, and $a_i = !(p,q,m_1)$, $a_j = ?(q,p,m_2)$:

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \;=\; \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)} \quad \text{implies} \quad m_1 = m_2$$

A proper word w is well-formed if $\sum_{m \in \mathcal{C}} |w|_{!(p,q,m)} = \sum_{m \in \mathcal{C}} |w|_{?(q,p,m)}$

Well-formedness and CFMs

Lemma

For any CFM $\mathcal{A}$ and $w \in Lin(\mathcal{A})$, w is well-formed.

Recall that there is a strong correspondence between well-formed linearizations and MSCs.

From linearizations to partial orders (reminder)

Associate to $w = a_1 \dots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

From linearizations to partial orders (reminder)

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$

From linearizations to partial orders (reminder)

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\prec_{\text{msg}} \cup \bigcup_{p \in \mathcal{P}} \prec_p \right)^*$ where

From linearizations to partial orders (reminder)

Associate to $w = a_1 \ldots a_n \in \mathit{Act}^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\prec_{\text{msg}} \ \cup \ \bigcup_{p \in \mathcal{P}} \prec_p \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$ for any $i, j \in E_p$

From linearizations to partial orders (reminder)

Associate to $w = a_1 \ldots a_n \in Act^*$ an Act-labelled poset

$$M(w) = (E, \preceq, \ell)$$

such that:

- $E = \{1, \ldots, n\}$ are the positions in w labelled with $\ell(i) = a_i$
- $\preceq = \left(\prec_{\text{msg}} \ \cup \ \bigcup_{p \in \mathcal{P}} \prec_p \right)^*$ where
  - $i \prec_p j$ if and only if $i < j$ for any $i, j \in E_p$
  - $i \prec_{\text{msg}} j$ if for some $(p, q) \in Ch$ and $m \in \mathcal{C}$ we have:

$$\ell(i) = !(p, q, m) \text{ and } \ell(j) = ?(q, p, m) \text{ and}$$

$$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|_{!(p,q,m)} \ = \ \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|_{?(q,p,m)}$$

CFMs and well-formed words

Relating well-formed words to MSCs

For any well-formed word $w \in Act^*$, $M(w)$ is an MSC.

CFMs and well-formed words

Relating well-formed words to MSCs

For any well-formed word $w \in Act^*$, $M(w)$ is an MSC.

Definition (MSC language of a CFM)

For CFM $\mathcal{A}$, let $\mathcal{L}(\mathcal{A}) = \{\, M(w) \mid w \in Lin(\mathcal{A}) \,\}$.

CFMs and well-formed words

Relating well-formed words to MSCs

For any well-formed word $w \in Act^*$, $M(w)$ is an MSC.

Definition (MSC language of a CFM)

For CFM $\mathcal{A}$, let $\mathcal{L}(\mathcal{A}) = \{ M(w) \mid w \in Lin(\mathcal{A}) \}$.

Relating well-formed words to CFMs

For any well-formed words u and v with $M(u)$ is isomorphic to $M(v)$:

$$\text{for any CFM } \mathcal{A}: \ u \in \mathcal{L}(\mathcal{A}) \quad \text{iff} \quad v \in \mathcal{L}(\mathcal{A}).$$

Overview

1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs

Emptiness problem is undecidable for CFMs

Theorem: [Brand & Zafiropulo 1983]

The following (emptiness) problem:

INPUT:	CFM $\mathcal{A}$ over processes $\mathcal{P}$ and message contents $\mathcal{C}$
QUESTION:	Is $\mathcal{L}(\mathcal{A})$ empty?

is undecidable.

Emptiness problem is undecidable for CFMs

Theorem: [Brand & Zafiropulo 1983]

The following (emptiness) problem:

INPUT:	CFM $\mathcal{A}$ over processes $\mathcal{P}$ and message contents $\mathcal{C}$
QUESTION:	Is $\mathcal{L}(\mathcal{A})$ empty?

is undecidable. (Even if $\mathcal{C}$ is a singleton set).

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- Consider restricted versions of CFMs by bounding the channel capacities.
- Thus: we fix the channel capacities a priori.

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- Consider restricted versions of CFMs by bounding the channel capacities.
- Thus: we fix the channel capacities a priori.
- This yields:

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- Consider restricted versions of CFMs by bounding the channel capacities.
- Thus: we fix the channel capacities a priori.
- This yields:
  - universally bounded CFMs: all runs need a finite buffer capacity
  - existentially bounded CFMs: some runs need a finite buffer capacity

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- Consider restricted versions of CFMs by bounding the channel capacities.
- Thus: we fix the channel capacities a priori.
- This yields:
  - universally bounded CFMs: all runs need a finite buffer capacity
  - existentially bounded CFMs: some runs need a finite buffer capacity possibly, some runs still need unbounded buffers.

Restrictions on CFMs

- So: most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- Consider restricted versions of CFMs by bounding the channel capacities.
- Thus: we fix the channel capacities a priori.
- This yields:
  - universally bounded CFMs: <u>all</u> runs need a finite buffer capacity
  - existentially bounded CFMs: <u>some</u> runs need a finite buffer capacity possibly, some runs still need unbounded buffers.

We define bounded CFMs, by first considering bounded words and bounded MSCs. Bounded CFMs will then generate bounded MSCs.

Bounded words

Definition (B-bounded words)

Let $B \in \mathbb{N}$ and $B > 0$. A word $w \in Act^*$ is called B-bounded if for any prefix u of w and any channel $(p, q) \in Ch$:

$$0 \leqslant \sum_{a \in \mathcal{C}} |u|_{!(p,q,a)} - \sum_{a \in \mathcal{C}} |u|_{?(q,p,a)} \leqslant B$$

Bounded words

Definition (B-bounded words)

Let $B \in \mathbb{N}$ and $B > 0$. A word $w \in Act^*$ is called B-bounded if for any prefix u of w and any channel $(p, q) \in Ch$:

$$0 \leqslant \sum_{a \in \mathcal{C}} |u|_{!(p,q,a)} - \sum_{a \in \mathcal{C}} |u|_{?(q,p,a)} \leqslant B$$

Intuition

Word w is B-bounded if for any pair of processes (p, q), the number of sends from p to q cannot be more than B ahead of the number of receipts by q from p (for every message a).

Bounded words

Definition (B-bounded words)

Let $B \in \mathbb{N}$ and $B > 0$. A word $w \in Act^*$ is called B-bounded if for any prefix u of w and any channel $(p, q) \in Ch$:

$$0 \leqslant \sum_{a \in \mathcal{C}} |u|_{!(p,q,a)} - \sum_{a \in \mathcal{C}} |u|_{?(q,p,a)} \leqslant B$$

Intuition

Word w is B-bounded if for any pair of processes (p, q), the number of sends from p to q cannot be more than B ahead of the number of receipts by q from p (for every message a).

Example

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

Bounded MSCs

Definition (Universally bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$.

Bounded MSCs

Definition (Universally bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called universally B-bounded ($\forall B$-bounded, for short) if

Bounded MSCs

Definition (Universally bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called universally B-bounded ($\forall B$-bounded, for short) if

$$Lin(M) \; = \; Lin^{B}(M)$$

where $Lin^{B}(M) := \{w \in Lin(M) \mid w \text{ is } B\text{-bounded}\}$.

Bounded MSCs

Definition (Universally bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called universally B-bounded ($\forall B$-bounded, for short) if

$$Lin(M) \;=\; Lin^{B}(M)$$

where $Lin^{B}(M) := \{w \in Lin(M) \mid w \text{ is } B\text{-bounded}\}$.

Intuition

MSC M is $\forall B$-bounded if all its linearizations are B-bounded.

Bounded MSCs

Definition (Universally bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called universally B-bounded ($\forall B$-bounded, for short) if

$$Lin(M) \; = \; Lin^{B}(M)$$

where $Lin^{B}(M) := \{w \in Lin(M) \mid w \text{ is } B\text{-bounded}\}$.

Intuition

MSC M is $\forall B$-bounded if all its linearizations are B-bounded.

So: if M is B-bounded, then a buffer capacity B is sufficient for all possible runs of MSC M.

Bounded MSCs

Definition (Existentially bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called existentially B-bounded ($\exists B$-bounded, for short) if $Lin(M) \cap Lin^B(M) \neq \varnothing$.

Bounded MSCs

Definition (Existentially bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called existentially B-bounded ($\exists B$-bounded, for short) if $Lin(M) \cap Lin^B(M) \neq \varnothing$.

Intuition

MSC M is $\exists B$-bounded if at least one linearization of M is B-bounded.

Bounded MSCs

Definition (Existentially bounded MSCs)

Let $B \in \mathbb{N}$ and $B > 0$. An MSC $M \in \mathbb{M}$ is called existentially B-bounded ($\exists B$-bounded, for short) if $Lin(M) \cap Lin^B(M) \neq \varnothing$.

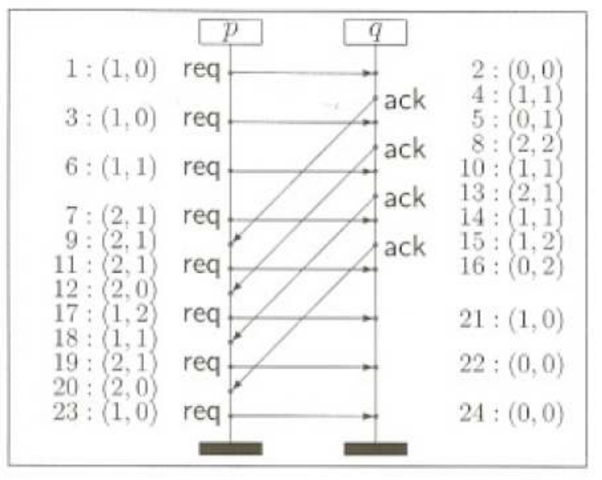
Intuition

MSC M is $\exists B$-bounded if at least one linearization of M is B-bounded.

Consequence

The MSC M can be "scheduled" in such a way that no channel ever contains more than B messages.

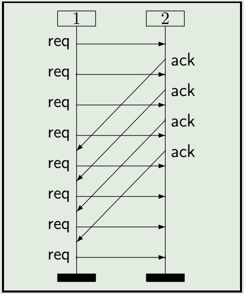
Bounded MSCs

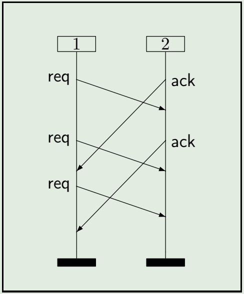


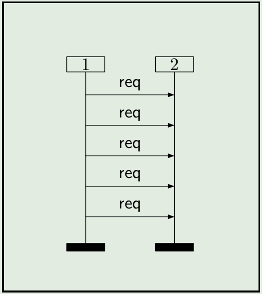
An ∃2-bounded MSC with a corresponding justification

Bounded MSCs

Example

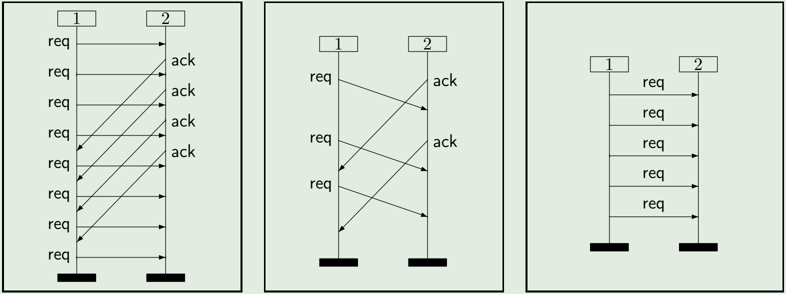






Bounded MSCs

Example



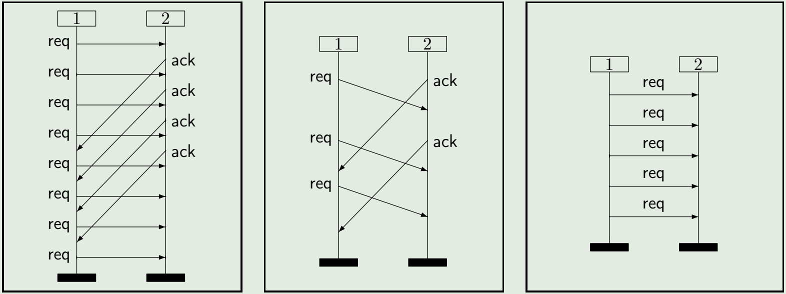
$\forall$4-bounded

$\exists$2-bounded

not $\exists$1-bounded

Bounded MSCs

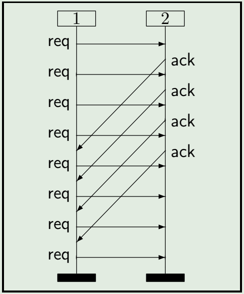
Example



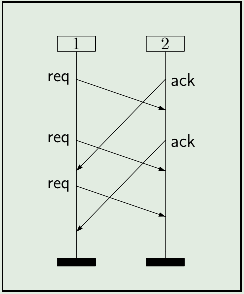
$\forall$4-bounded	$\forall$3-bounded
$\exists$2-bounded	$\exists$1-bounded
not $\exists$1-bounded	

Bounded MSCs

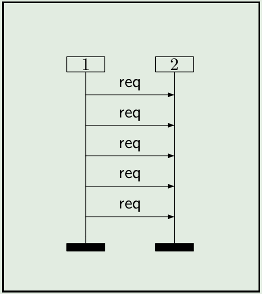
Example



$\forall$4-bounded
$\exists$2-bounded
not $\exists$1-bounded



$\forall$3-bounded
$\exists$1-bounded



$\forall$5-bounded
$\exists$1-bounded

Bounded CFMs

Definition (Universally bounded CFM)

1. Let $B \in \mathbb{N}$ and $B > 0$. CFM $\mathcal{A}$ is *universally B-bounded* if each MSC in $\mathcal{L}(\mathcal{A})$ is $\forall B$-bounded.
2. CFM $\mathcal{A}$ is *universally bounded* if it is $\forall B$-bounded for some $B \in \mathbb{N}$ and $B > 0$.

Bounded CFMs

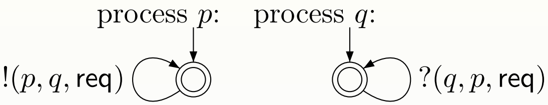
Definition (Universally bounded CFM)

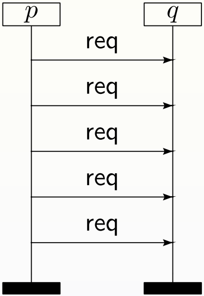
1. Let $B \in \mathbb{N}$ and $B > 0$. CFM $\mathcal{A}$ is *universally B-bounded* if each MSC in $\mathcal{L}(\mathcal{A})$ is $\forall B$-bounded.
2. CFM $\mathcal{A}$ is *universally bounded* if it is $\forall B$-bounded for some $B \in \mathbb{N}$ and $B > 0$.

Definition (Existentially bounded CFM)

1. Let $B \in \mathbb{N}$ and $B > 0$. CFM $\mathcal{A}$ is *existentially B-bounded* if each MSC in $\mathcal{L}(\mathcal{A})$ is $\exists B$-bounded.
2. CFM $\mathcal{A}$ is *existentially bounded* if it is $\exists B$-bounded for some $B \in \mathbb{N}$ and $B > 0$.

Example (1)



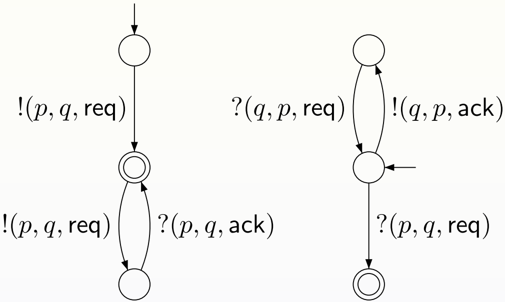


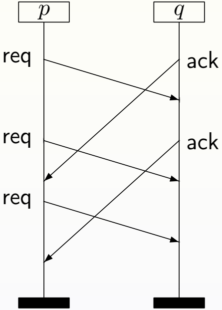
$\exists$1-bounded, but not $\forall B$-bounded for any B

so, not $\forall$-bounded.

Example (2)

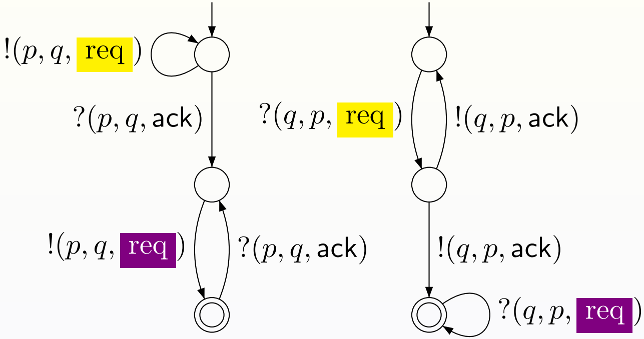
process p: process q:





∃1-bounded, and ∀3-bounded

Example (3)



$\exists\lceil\frac{n}{2}\rceil$-bounded, but not $\forall B$-bounded for any B

Justification

- Phase 1: process p sends n messages to q
  - messages of phase 1 are tagged with data req

- … and waits for the first acknowledgement of q

Justification

- Phase 1: process p sends n messages to q
  - messages of phase 1 are tagged with data req

- ... and waits for the first acknowledgement of q

- Phase 2: each ack is directly answered by p by another message
  - messages of phase 2 are tagged with data req

Justification

- Phase 1: process p sends n messages to q
  - messages of phase 1 are tagged with data req
- … and waits for the first acknowledgement of q
- Phase 2: each ack is directly answered by p by another message
  - messages of phase 2 are tagged with data req
- So, p sends $2n$ reqs to q and q sends n acks
  - existentially $\lceil \frac{n}{2} \rceil$-bounded
  - q starts to send acks after $\lceil \frac{n}{2} \rceil$ requests have been sent by p
  - after n sends, process p receives the first ack; then phase 2 starts
  - in phase 2, process p and q keep sending and receiving messages "in sync"
- Note: the CFM is also non-deterministic, and may deadlock.

Justification

- Phase 1: process p sends n messages to q
  - messages of phase 1 are tagged with data req
- ... and waits for the first acknowledgement of q
- Phase 2: each ack is directly answered by p by another message
  - messages of phase 2 are tagged with data req
- So, p sends $2n$ reqs to q and q sends n acks
  - existentially $\lceil \frac{n}{2} \rceil$-bounded
  - q starts to send acks after $\lceil \frac{n}{2} \rceil$ requests have been sent by p
  - after n sends, process p receives the first ack; then phase 2 starts
  - in phase 2, process p and q keep sending and receiving messages "in sync"
- Note: the CFM is also non-deterministic, and may deadlock. Why?

Emptiness is decidable for ∃-bounded CFMs

Theorem: [Genest *et. al*, 2006]

For any ∃-bounded CFM, the emptiness problem is decidable (and is PSPACE-complete).

Emptiness is decidable for ∃-bounded CFMs

Theorem: [Genest *et. al*, 2006]

For any ∃-bounded CFM, the emptiness problem is decidable (and is PSPACE-complete).

Note:

This decision problem is undecidable for arbitrary CFM, and is obviously decidable for ∀-bounded CFMs, as ∀-bounded CFMs have finitely many configurations, and thus one can check whether a configuration (s, η_ε) with $s \in F$ is reachable by a simple graph analysis.

Some (un)decidability results

Some (un)decidability results

Undecidable

The following problems on CFM $\mathcal{A}$ are all undecidable:

1. Is CFM $\mathcal{A}$ universally bounded?

Some (un)decidability results

Undecidable

The following problems on CFM $\mathcal{A}$ are all undecidable:

1. Is CFM $\mathcal{A}$ universally bounded?
2. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\forall B$-bounded?

Some (un)decidability results

Undecidable

The following problems on CFM $\mathcal{A}$ are all undecidable:

1. Is CFM $\mathcal{A}$ universally bounded?
2. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\forall B$-bounded?
3. Is CFM $\mathcal{A}$ existentially bounded?

Some (un)decidability results

Undecidable

The following problems on CFM $\mathcal{A}$ are all undecidable:

1. Is CFM $\mathcal{A}$ universally bounded?
2. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\forall B$-bounded?
3. Is CFM $\mathcal{A}$ existentially bounded?
4. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\exists B$-bounded?

Some (un)decidability results

Undecidable

The following problems on CFM $\mathcal{A}$ are all undecidable:

1. Is CFM $\mathcal{A}$ universally bounded?
2. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\forall B$-bounded?
3. Is CFM $\mathcal{A}$ existentially bounded?
4. For $B \in \mathbb{N}$ and $B > 0$, is CFM $\mathcal{A}$ $\exists B$-bounded?

the proofs of all these facts are left as an exercise

Deadlocks

Deadlocks

Deadlock-free CFMs

$(\overline{s}, \eta) \in Conf_{\mathcal{A}}$ is a deadlock configuration of CFM $\mathcal{A}$ if there is no 'accepting" configuration $(\overline{s}', \eta') \in F \times \{\eta_\varepsilon\}$ with $(\overline{s}, \eta) \Longrightarrow^*_{\mathcal{A}} (\overline{s}', \eta')$.

Deadlocks

Deadlock-free CFMs

$(\overline{s}, \eta) \in Conf_{\mathcal{A}}$ is a deadlock configuration of CFM $\mathcal{A}$ if there is no 'accepting" configuration $(\overline{s}', \eta') \in F \times \{\eta_\varepsilon\}$ with $(\overline{s}, \eta) \Longrightarrow^*_{\mathcal{A}} (\overline{s}', \eta')$.

CFM $\mathcal{A}$ is deadlock-free whenever it has no reachable deadlock configuration.

Deadlocks

Deadlock-free CFMs

$(\overline{s}, \eta) \in Conf_{\mathcal{A}}$ is a deadlock configuration of CFM $\mathcal{A}$ if there is no 'accepting" configuration $(\overline{s}', \eta') \in F \times \{\eta_\varepsilon\}$ with $(\overline{s}, \eta) \Longrightarrow^*_{\mathcal{A}} (\overline{s}', \eta')$.

CFM $\mathcal{A}$ is deadlock-free whenever it has no reachable deadlock configuration.

Checking deadlock-freeness is undecidable

The decision problem: Is CFM $\mathcal{A}$ deadlock free? is undecidable.

Deadlocks

Deadlock-free CFMs

$(\overline{s}, \eta) \in Conf_{\mathcal{A}}$ is a deadlock configuration of CFM $\mathcal{A}$ if there is no 'accepting" configuration $(\overline{s}', \eta') \in F \times \{\eta_\varepsilon\}$ with $(\overline{s}, \eta) \Longrightarrow^*_{\mathcal{A}} (\overline{s}', \eta')$.

CFM $\mathcal{A}$ is deadlock-free whenever it has no reachable deadlock configuration.

Checking deadlock-freeness is undecidable

The decision problem: Is CFM $\mathcal{A}$ deadlock free? is undecidable.

Checking B-boundedness for deadlock-free CFMs is decidable

The decision problem: for deadlock-free CFM $\mathcal{A}$ and $B \in \mathbb{N}$ with $B > 0$, is $\mathcal{A}$ $\forall B$-bounded? is decidable.