Theoretical Foundations of the UML
Lecture 8: Bounded MSCs and CFMs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

moves.rwth-aachen.de/teaching/ss-20/fuml/

May 12, 2020
1. Communicating finite-state machines: a refresher

2. Well-formedness of CFMs

3. Bounded CFMs
   - Bounded words ✓
   - Bounded MSCs ✓
   - Bounded CFMs ✓

Well-formed language of words

Well-formed language of MSCs

Bounded = "bound the capacity of the communication channels of a CFM"
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 $A_p$ and $A_q$ are in some local state, and (as usual) all channels are empty.
A communicating finite-state machine (CFM) over $\mathcal{P}$ and $\mathcal{C}$ is a tuple $A = (((S_p, \Delta_p))_{p \in \mathcal{P}}, D, s_{init}, F)$ 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

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

where

- for each \( p \in \mathcal{P} \):
  - \( S_p \) is a non-empty finite set of local states (the \( S_p \) are disjoint)
  - \( \Delta_p \subseteq S_p \times \text{Act}_p \times \mathbb{D} \times S_p \) is a set of local transitions

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

\[ A = (((S_p, \Delta_p))_{p \in P}, D, s_{init}, F) \]

where

- for each \( p \in P \):
  - \( S_p \) is a non-empty finite set of *local states* (the \( S_p \) are disjoint)
  - \( \Delta_p \subseteq S_p \times Act_p \times D \times S_p \) is a set of *local transitions*
- \( D \) is a nonempty finite set of *synchronization messages* (or *data*)
- \( s_{init} \in S_A \) is the *global initial state*
- where \( S_A := \prod_{p \in P} S_p \) is the set of *global states* of \( A \)

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

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

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

where

- for each \( p \in \mathcal{P} \):
  - \( S_p \) is a non-empty finite set of **local states** (the \( S_p \) are disjoint)
  - \( \Delta_p \subseteq S_p \times Act_p \times \mathcal{D} \times S_p \) is a set of **local transitions**
- \( \mathcal{D} \) 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 \( \mathcal{A} = (((S_p, \Delta_p))_{p \in \mathcal{P}}, \mathcal{D}, s_{init}, F) \) be a CFM over \( \mathcal{P} \) and \( \mathcal{C} \).
Formal semantics of CFMs

Definition (Configuration)

Configurations of $A$: $Conf_A := S_A \times \{ \eta \mid \eta : Ch \rightarrow (C \times D)^* \}$

- Global state
- Context of all channels

Let $p$ be a process and $g$ be a global content of all state channels.

$$\eta(p,g) = (\text{ack,R})(\text{ack,L})(\text{reg,R})(\text{reg,L})$$
Formal semantics of CFMs

**Definition (Configuration)**

Configurations of $A$: $\text{Conf}_A := S_A \times \{\eta \mid \eta : \text{Ch} \rightarrow (C \times D)^*\}$

**Definition (Transitions between configurations)**

$\Rightarrow_A \subseteq \text{Conf}_A \times \text{Act} \times D \times \text{Conf}_A$ is defined as follows:

- Sending a message: \((\langle s, \eta \rangle, !(p, q, a), m, \langle s', \eta' \rangle) \in \Rightarrow_A \text{ if} \)
  \[\begin{align*}
  (s[p], !(p, q, a), m, s'[p]) &\in \Delta_p \\
  \eta' &= \eta[(p, q) := (a, m) \cdot \eta((p, q))] \\
  s[r] &= s'[r] \text{ for all } r \in P \setminus \{p\}
  \end{align*}\]

\[\begin{align*}
\tau &= (\bar{s}, \eta) \\
\downarrow !(r_{eq}, L) \\
\tau' &= (\bar{s}', \eta')
\end{align*}\]
Formal semantics of CFMs

Definition (Configuration)

Configurations of $A$: $\text{Conf}_A := S_A \times \{\eta \mid \eta : \text{Ch} \to (C \times D)^*\}$

Definition (Transitions between configurations)

$\xRightarrow{A} \subseteq \text{Conf}_A \times \text{Act} \times D \times \text{Conf}_A$ is defined as follows:

- sending a message: $((\bar{s}, \eta), !(p, q, a), m, (\bar{s}', \eta')) \in \xRightarrow{A}$ if
  - $(\bar{s}[p], !(p, q, a), m, \bar{s}'[p]) \in \Delta_p$
  - $\eta' = \eta[(p, q) := (a, m) \cdot \eta((p, q))]$
  - $\bar{s}[r] = \bar{s}'[r]$ for all $r \in P \setminus \{p\}$

- receipt of a message: $((\bar{s}, \eta), ?(p, q, a), m, (\bar{s}', \eta')) \in \xRightarrow{A}$ if
  - $(\bar{s}[p], ?(p, q, a), m, \bar{s}'[p]) \in \Delta_p$
  - $\eta((q, p)) = w \cdot (a, m) \neq \epsilon$ and $\eta' = \eta[(q, p) := w]$
  - $\bar{s}[r] = \bar{s}'[r]$ for all $r \in P \setminus \{p\}$
Definition ((Accepting) Runs)

A run of $A$ 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

- $\gamma_0 = (s_{\text{init}}, \eta_\varepsilon)$ with $\eta_\varepsilon$ mapping any channel to $\varepsilon$
- $\gamma_{i-1} \xrightarrow{\sigma_i, m_i} A \gamma_i$ for any $i \in \{1, \ldots, n\}$
A run of $\mathcal{A}$ on $\sigma_1 \ldots \sigma_n \in \text{Act}^*$ is a sequence $\rho = \gamma_0 \: m_1 \: \gamma_1 \: \ldots \: \gamma_{n-1} \: m_n \: \gamma_n$ such that

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

Run $\rho$ is accepting if $\gamma_n \in F \times \{\eta_\varepsilon\}$.
Definition ((Accepting) Runs)

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

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

Run $\rho$ is accepting if $\gamma_n \in F \times \{\eta_\epsilon\}$.

Definition (Linearizations)

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

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

This CFM accepts if $A_p$ and $A_q$ are in some local state, and (as usual) all channels are empty.
Initial configuration:

\[ \left( (1, A), \varepsilon, \varepsilon \right) \xrightarrow{!\ (\text{req}, R)} \left( (3, A), (\text{req}, R), \varepsilon \right) \xrightarrow{?\ (\text{req}, R)} \left( (3, C), \varepsilon, \varepsilon \right) \xrightarrow{!\ (\text{ack}, L)} \left( (3, B), \varepsilon, (\text{ack}, L) \right) \]

Non-accepting run:

- Initial state: \( (1, A), \varepsilon, \varepsilon \)
- Transition: \( !\ (\text{req}, R) \)
- New state: \( (3, A), (\text{req}, R), \varepsilon \)
- Transition: \( ?\ (\text{req}, R) \)
- New state: \( (3, C), \varepsilon, \varepsilon \)
- Transition: \( !\ (\text{ack}, L) \)
- New state: \( (3, B), \varepsilon, (\text{ack}, L) \)
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

\begin{itemize}
  \item every receive in $w$ is preceded by a corresponding send, i.e.:
    \begin{itemize}
      \item $\forall (p, q) \in Ch$ and prefix $u$ of $w$, we have:
        \begin{align*}
          \sum_{m \in \mathcal{C}} |u|!(p, q, m) & \geq \sum_{m \in \mathcal{C}} |u|?(q, p, m)
        \end{align*}
    \end{itemize}
\end{itemize}

# sends from $p$ to $q$ # receipts by $q$ 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 \( w = a_1 \ldots a_n \in \text{Act}^* \) proper if:

1. every receive in \( w \) is preceded by a corresponding send, i.e.:
   \[ \forall (p, q) \in Ch \text{ and prefix } u \text{ of } w, \text{ we have:} \]
   \[
   \sum_{m \in \mathcal{C}} |u| !_{(p,q,m)} \geq \sum_{m \in \mathcal{C}} |u| ?_{(q,p,m)}
   \]
   \( \sum \) sends from \( p \) to \( q \) \( \sum \) receipts by \( q \) from \( p \)

   where \( |u|_a \) denotes the number of occurrences of action \( a \) in \( u \)

2. the FIFO policy is respected, i.e.:
   \[ \forall 1 \leq i < j \leq n, (p, q) \in Ch, \text{ and } a_i = !(p, q, m_1), a_j = ?(q, p, m_2): \]
   \[
   \sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}| !_{(p,q,m)} = \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}| ?_{(q,p,m)} \implies m_1 = m_2
   \]
   \( \sum \) sends from \( p \) to \( q \) \( \sum \) receives at \( q \) 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 $w = a_1 \ldots a_n \in \text{Act}^*$ 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:
   \[
   \sum_{m \in C} |u|!(p, q, m) \geq \sum_{m \in C} |u|?(q, p, m)
   \]
   \[
   \text{# sends from } p \text{ to } q \quad \text{# receipts by } q \text{ from } p
   \]
   where $|u|_a$ denotes the number of occurrences of action $a$ in $u$

2. the FIFO policy is respected, i.e.:
   $\forall 1 \leq i < j \leq n$, $(p, q) \in Ch$, and $a_i = !(p, q, m_1)$, $a_j = ?(q, p, m_2)$:
   \[
   \sum_{m \in C} |a_1 \ldots a_{i-1}!(p, q, m)| = \sum_{m \in C} |a_1 \ldots a_{j-1}?(q, p, m)| \quad \text{implies} \quad m_1 = m_2
   \]

A proper word $w$ is well-formed if $\sum_{m \in C} |w|!(p, q, m) = \sum_{m \in C} |w|?(q, p, m)$.
Well-formedness and CFMs

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

Recall that there is a strong correspondence between well-formed linearizations and MSCs.
Associate to $w = a_1 \ldots a_n \in Act^*$ an $Act$-labelled poset

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

such that:

$$\ell : E \rightarrow Act$$
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 $\ell(i) = a_i$
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 $\ell(i) = a_i$
- $\preceq = (\prec_{\text{msg}} \cup \bigcup_{p \in \mathcal{P}} \prec_p)^*$ where...
From linearizations to partial orders (reminder)

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

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

such that:

- $E = \{1, \ldots, n\}$ are the positions in $w$ labelled with $\ell(i) = a_i$
- $\preceq = (\prec_{\text{msg}} \cup \bigcup_{p \in \mathcal{P}} \prec_p)^*$ 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 $w = a_1 \ldots a_n \in \text{Act}^*$ an $\text{Act}$-labelled poset

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

such that:

- $E = \{1, \ldots, n\}$ are the positions in $w$ labelled with $\ell(i) = a_i$
- $\preceq = (\prec_{\text{msg}} \cup \bigcup_{p \in \mathcal{P}} \prec_p)^*$ 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 $(p, q) \in \mathcal{Ch}$ and $m \in \mathcal{C}$ we have:
    $$\ell(i) = !(p, q, m) \quad \text{and} \quad \ell(j) = ?(q, p, m)$$
    $$\sum_{m \in \mathcal{C}} |a_1 \ldots a_{i-1}|!(p, q, m) = \sum_{m \in \mathcal{C}} |a_1 \ldots a_{j-1}|?(q, p, m)$$
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 \( w \in \text{Act}^* \), \( M(w) \) is an MSC.

Definition (MSC language of a CFM)
For CFM \( A \), let
\[
\mathcal{L}(A) = \{ M(w) \mid w \in \text{Lin}(A) \}.
\]
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 $A$, let $\mathcal{L}(A) = \{ M(w) \mid w \in Lin(A) \}$.

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

for any CFM $A$ : $u \in \mathcal{L}(A)$ iff $v \in \mathcal{L}(A)$.
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 $A$ over processes $\mathcal{P}$ and message contents $\mathcal{C}$

**QUESTION:** Is $\mathcal{L}(A)$ empty?

is **undecidable**.
Theorem: [Brand & Zafiropulo 1983]

The following (emptiness) problem:

**INPUT:** CFM $A$ over processes $\mathcal{P}$ and message contents $\mathcal{C}$

**QUESTION:** Is $\mathcal{L}(A)$ empty?

is undecidable. (Even if $\mathcal{C}$ is a singleton set).
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:
  - universally bounded CFMs: all runs need a finite buffer capacity.
  - existentially bounded CFMs: some runs need a finite buffer capacity possibly, some runs may still need unbounded buffers.

We define bounded CFMs, by first considering bounded words and bounded MSCs. Bounded CFMs will then "generate" bounded MSCs. a.k.a. accept
Definition ($B$-bounded words)

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

$$0 \leq \sum_{a \in C} |u|!(p, q, a) - \sum_{a \in C} |u|?(q, p, a) \leq B$$
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 $(p, q) \in Ch$:

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

**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 $w \in \text{Act}^*$ is called $B$-bounded if for any prefix $u$ of $w$ and any channel $(p, q) \in \text{Ch}$:

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

**(+)\)**

**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.
Example

\[ w = \frac{! (p, g, a)}{1} \frac{! (p, g, a)}{1} \frac{(a, p, a)}{2} \frac{! (p, g, a)}{1} \frac{! (a, p, b)}{2} \frac{(a, p, a)}{1} \]

**Claim:** \( w \) is 3-bounded  \[ B = 3 \]

- \( \varepsilon = 1 \)
- \( ! (p, g, a) = 1 \)
- \( ! (p, g, a) \ 1 \frac{(p, g, a)}{2} \frac{(p, g, a)}{2} \) = 2
- \( ! (p, g, a) \ 1 (a, p, a) = 1 \)

**Claim:** \( w \) is 2-bounded

**Proposition**

if \( w \) is \( B \)-bounded

then \( w \) is \( (B+1) \)-bounded

Typically, we are interested in determining the minimal \( B \) for which \( w \) is \( B \)-bounded.
Bounded MSCs

Definition (Universally bounded MSCs)

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

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

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

Joost-Pieter Katoen
Theoretical Foundations of the UML
**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

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

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

**Intuition**

MSC $M$ is $\forall B$-bounded if all its linearizations are $B$-bounded.
Claim MSC $M$ is $\mathcal{A}_3$-banded

$\mathcal{A}_2$-banded

$\mathcal{A}_3$-banded

$\text{Lin}(M) = \{ 1, 2, 3, 4, 5, 8, 7, 10, g \}$

2-banded

3-banded

$\begin{array}{cccccccccccc}
(p,q) & 0 & 1 & 1 & 0 & 1 & 1 & 1 & 1 & 2 & 1 & 1 & 0 \\
0 & 1 & 2 & 3 & 4 & 6 & 5 & 8 & 7 & 10 & g \\
0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0 & 0 \\
\end{array}$

$\begin{array}{cccccccccccc}
(p,q) & 0 & 1 & 2 & 2 & 2 & 3 & 2 & 2 & 2 & 2 & 1 & 0 \\
0 & 1 & 4 & 2 & 6 & 8 & 3 & 5 & 10 & 7 & 9 \\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
\end{array}$
Bounded MSCs

Definition (Universally bounded MSCs)

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

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

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

Intuition

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

So: if $M$ is $\forall B$-bounded, then a buffer capacity $B$ is sufficient for all possible runs of MSC $M$. 
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 $\text{Lin}(M) \cap \text{Lin}^B(M) \neq \emptyset$. 

Bounded MSCs
Proposition

if MSC M is FB-bonded,
then it is $F(B+1)$-bonded.

Similarly for $F_1B$-bonded MSCs.
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* (∃$B$-bounded, for short) if $Lin(M) \cap Lin^B(M) \neq \emptyset$.

**Intuition**

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

**Consequence**

The events of an ∃$B$-bounded MSC $M$ can be “scheduled” in such a way that no channel ever contains more than $B$ messages.
Bounded MSCs

An \( \exists 2 \)-bounded MSC with a corresponding justification

Find a "schedule" in which messages are received as soon as possible.
Bounded MSCs

Example

\[\begin{array}{c}
1 \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\hline
2 \\
\text{ack} \\
\text{ack} \\
\text{ack} \\
\text{ack} \\
\end{array}\]

\[\begin{array}{c}
1 \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\hline
2 \\
\text{ack} \\
\text{ack} \\
\end{array}\]

\[\begin{array}{c}
1 \\
\text{req} \\
\text{req} \\
\text{req} \\
\text{req} \\
\hline
2 \\
\end{array}\]

(*)
what is the minimal \( B \) s.t. \( M \) is \( FB \)-bounded?

\[ B = 1 \]

\[
\begin{array}{llllll}
(1,0) & (1,1) & (0,1) & (1,1) & (0,1) & (0,0) \\
1 & 2 & 3 & 4 & 5 & 6 \\
\text{req} & \text{ack} & \text{ack} & \text{req} & \text{req} & \text{req} \\
1 & 2 & 3 & 4 & 5 & 6 \\
\end{array}
\]

“Scheduling” strategy of the events is:
- receive messages as soon as possible
- postpone sending messages as long as possible
What is the minimal $B$ s.t.

\[ M \text{ is } \forall B\text{-bounded} ? \]

\[ B = 3 \]

\[(1, 0)\]
\[(2, 1)\]
\[(2, 0)\]
\[(3, 0)\]
\[(2, 0)\]

"Scheduling" strategy:

- defer receiving messages as long as possible
- send messages as soon as possible

\[ \Rightarrow \text{in order to find the maximal capacity needed for all possible linearisations} \]
Bounded MSCs

Example

∀4-bounded
∃2-bounded
not ∃1-bounded
The diagram illustrates a protocol with two nodes labeled 1 and 2. The protocol starts with a request (req) at node 1 and an acknowledgment (ack) at node 2. The sequence continues with further requests and acknowledgments:

1. (1.0) req
2. (1.0) ack
3. (1.0) req
4. (1.0) ack
5. (1.0) req
6. (1.0) ack
7. (1.0) req
8. (1.0) ack

The protocol continues with a series of requests and acknowledgments, labeled as 9, 10, etc., alternating between nodes 1 and 2.

The text on the right side of the diagram notes that this is an example of an infinite sequence (\(\mathbb{Z}_2\)-bounded).
Bounded MSCs

Example

∀4-bounded
∃2-bounded
not ∃1-bounded

∀3-bounded
∃1-bounded

∀5-bounded
∃1-bounded
Bounded CFMs

**Definition (Universally bounded CFM)**

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

**Proposition** every $\forall B$-banded CFM has finitely many configurations.

(emptiness problem for $\forall B$-banded CFMs is obviously decidable)
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)

∃1-bounded, but not ∀B-bounded for any B so, not ∀-bounded.
Example (2)

∀3-bounded, and ∃1-bounded
Example (3)

∃⌈n/2⌉-bounded, but not ∀B-bounded for any B

p can send arbitrarily many messages to q

n = # yellow messages
Phase 1: process $p$ sends $n$ messages to $q$
- messages of phase 1 are tagged with data $\text{req}$

... 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 $\text{req}$

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.
Phase 1: process $p$ sends $n$ messages to $q$
- messages of phase 1 are tagged with data \texttt{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 \texttt{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 $\exists$-bounded CFMs

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

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

$\text{L}(A) = \emptyset$?
Emptiness is decidable for $\exists$-bounded CFMs

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

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

**Note:**

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

Undecidable

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

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

the proofs of all these facts are left as an exercise
Deadlocks

Deadlock-free CFMs

$(\vec{s}, \eta) \in Conf_A$ is a deadlock configuration of CFM $A$ if there is no “accepting” configuration $(\vec{s}', \eta') \in F \times \{\eta_{\varepsilon}\}$ with $(\vec{s}, \eta) \xrightarrow{A}^* (\vec{s}', \eta')$.

CFM $A$ is deadlock-free whenever it has no reachable deadlock configuration.

Checking deadlock-freeness is undecidable

The decision problem: Is CFM $A$ deadlock free? is undecidable.

Checking $B$-boundedness for deadlock-free CFMs is decidable

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