Theoretical Foundations of the UML

Lecture 7: Languages and Subclasses of CFMs

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

http://moves.rwth-aachen.de/i2/uml09100/

16. November 2012

RWTH AACHEN UNIVERSITY

Outline

1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs
4. Properties of CFMs
   - Deterministic CFMs
   - Deadlock-free CFMs
   - Synchronisation messages add expressiveness

Overview

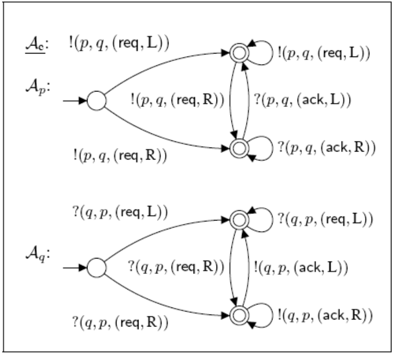
1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs
4. Properties of CFMs
   - Deterministic CFMs
   - Deadlock-free CFMs
   - Synchronisation messages add expressiveness

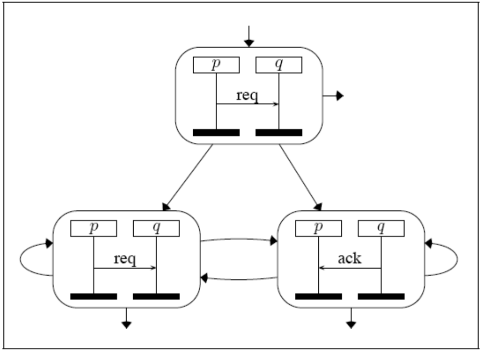
RWTH AACHEN UNIVERSITY

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





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

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

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
4. Properties of CFMs
   - Deterministic CFMs
   - Deadlock-free CFMs
   - Synchronisation messages add expressiveness

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

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

such that:

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

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

RWTH AACHEN UNIVERSITY

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

Relating well-formed words to CFMs

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

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

Overview

1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs
4. Properties of CFMs
   - Deterministic CFMs
   - Deadlock-free CFMs
   - Synchronisation messages add expressiveness

Emptiness problem is undecidable for CFMs

Theorem: [Brand & Zafiropulo 1983]

The following problem:

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

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

RWTH AACHEN UNIVERSITY

Restrictions on CFMs

- So: the emptiness problem is undecidable.
- Thus, most elementary problems for CFMs are undecidable.
- This is (very) unsatisfactory.
- Main cause: presence of channels with unbounded capacity
- We will therefore consider a restricted version of CFMs:
  - Consider bounded channels. This yields:
  - universally bounded CFMs: all runs need a finite buffer capacity
  - existentially bounded CFMs: some runs need a finite buffer capacity

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 $(p, q) \in Ch$:

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

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 $B > 0$. 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 $Lin(M) \cap Lin^B(M) \neq \varnothing$.

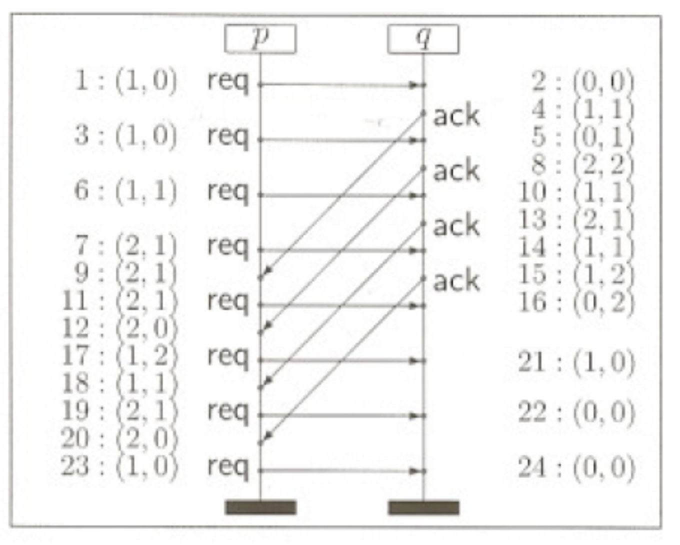
Intuition

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

Consequence

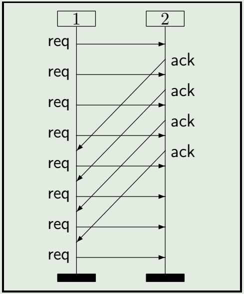
The MSC M can be "scheduled" in such a way that none of the channels ever contains more than B messages.

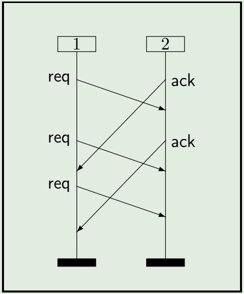
Bounded MSCs

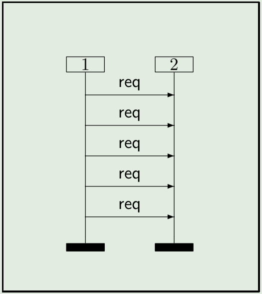


Bounded MSCs

Example







∀4-bounded
∃2-bounded
not ∃1-bounded

∀3-bounded
∃1-bounded

∀5-bounded
∃1-bounded

RWTH AACHEN UNIVERSITY

Bounded CFMs

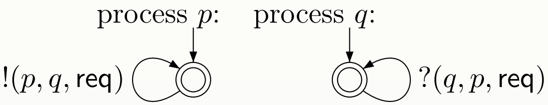
Definition (Universally bounded CFM)

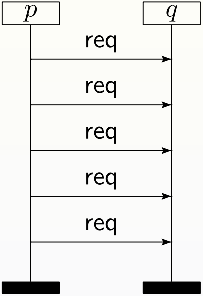
1. Let $B \in \mathbb{N}$ and $B > 0$. CFM $\mathcal{A}$ is *universally B-bounded* if any MSC in $L(\mathcal{A})$ is $\forall B$-bounded.
2. CFM $\mathcal{A}$ is *universally bounded* if it is $\forall B$-bounded for some $B \in \mathbb{N}$ and $B > 0$.

Definition (Existentially bounded CFM)

Let $B \in \mathbb{N}$ and $B > 0$. CFM $\mathcal{A}$ is *existentially B-bounded* if any MSC in $L(\mathcal{A})$ is $\exists B$-bounded.

Example (1)

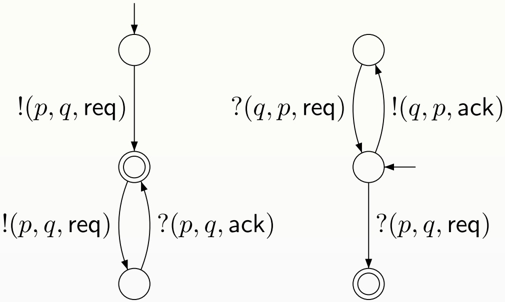


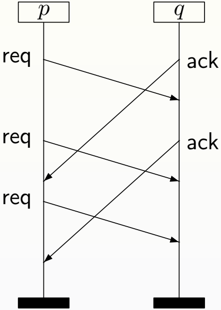


existentially 1-bounded, but not $\forall B$-bounded for any B

Example (2)

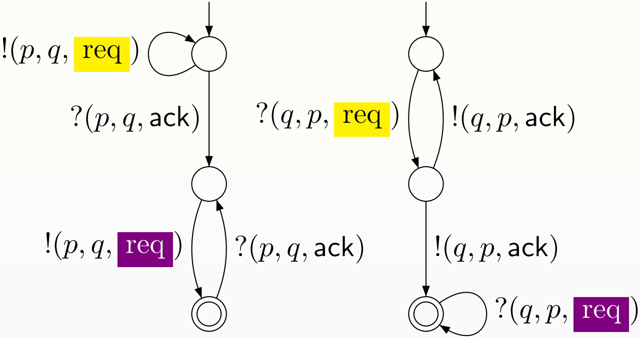
process p: process q:





existentially 1-bounded, and ∀3-bounded

Example (3)



existentially $\lceil \frac{n}{2} \rceil$-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
- 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, but not $\forall$-bounded
- The CFM is also non-deterministic, and may deadlock

RWTH AACHEN UNIVERSITY

Existentially bounded CFMs

Theorem: [Genest et. al, 2006]

For any $\exists B$-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 $\forall$-bounded CFMs, as they have finitely many configurations.

Overview

1. Communicating finite-state machines: a refresher
2. Well-formedness of CFMs
3. Bounded CFMs
   - Bounded words
   - Bounded MSCs
   - Bounded CFMs
4. Properties of CFMs
   - Deterministic CFMs
   - Deadlock-free CFMs
   - Synchronisation messages add expressiveness

Determinism

Definition (Deterministic CFM)

A CFM $\mathcal{A}$ is *deterministic* if for all $p \in \mathcal{P}$, the transition relation Δ_p satisfies the following two conditions:

1. $(s, !(p, q, (a, m_1)), s_1) \in \Delta_p$ and $(s, !(p, q, (a, m_2)), s_2) \in \Delta_p$ implies $m_1 = m_2$ and $s_1 = s_2$
2. $(s, ?(p, q, (m, \lambda)), s_1) \in \Delta_p$ and $(s, ?(p, q, (m, \lambda)), s_2) \in \Delta_p$ implies $s_1 = s_2$

Note:

From the same state, process p may have the possibility of sending messages to more than one process.

Example:

Example CFM (1) and (2) are deterministic, while (3) is not.

Deadlock-freeness

Definition (Deadlock-free CFM)

A CFM $\mathcal{A}$ is *deadlock-free* if, for all $w \in Act^*$ and all runs γ of $\mathcal{A}$ on w, there exist $w' \in Act^*$ and run γ' in $\mathcal{A}$ such that $\gamma \cdot \gamma'$ is an accepting run of $\mathcal{A}$ on $w \cdot w'$.

Example:

Example CFM (1) and (2) are deadlock-free, while (3) is not.

Theorem: [Genest et. al, 2006]

For any $\exists B$-bounded CFM $\mathcal{A}$, the decision problem "is $\mathcal{A}$ deadlock-free" is decidable (and is PSPACE-complete).

Weak CFMs

Definition (Weak CFM)

A CFM is called a *weak* CFM if $|\mathbb{D}| = 1$.

Are CFMs more expressive than weak CFMs? That is, do there exist languages (over linearizations or, equivalently, MSCs) that can be generated by CFMs but not by weak CFMs? Yes.

CFM vs. weak CFM

Theorem:

Weak CFMs are less expressive than CFMs.

Proof.

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

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

Claim: there is no weak CFM over $\{1,2\}$ and $\{\text{req}, \text{ack}\}$ whose language is $L = \{M(n,n) \mid n > 0\}$. By contraposition. Suppose there is a weak CFM $\mathcal{A} = ((\mathcal{A}_1, \mathcal{A}_2), \mathbb{D}, s_{init}, F)$ with $L(\mathcal{A}) = L$. For any $n > 0$, there is an accepting run of $\mathcal{A}$ on $M(n,n)$. If n is sufficiently large, then

- $\mathcal{A}_1$ visits a cycle of length $i > 0$ to read the first n letters of $M(n,n) \restriction 1$
- $\mathcal{A}_2$ visits a cycle of length $j > 0$ to read the last n letters of $M(n,n) \restriction 2$

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