# Theoretical Foundations of the UML

## Lecture 11: Safe Realisability


Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group


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

7. Juni 2016

# Outline

# Overview

1. Safe realisability
2. Closure and inference revisited
3. Characterisation and complexity of safe realisability

# From requirements to implementation

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

# From requirements to implementation

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

Questions:

1. Is this possible? (That is, is this decidable?)

# From requirements to implementation

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

Questions:

1. Is this possible? (That is, is this decidable?)
2. If so, how complex is it to obtain such CFM?

# From requirements to implementation

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

Questions:

1. Is this possible? (That is, is this decidable?)
2. If so, how complex is it to obtain such CFM?
3. If so, how do such algorithms work?

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different forms of requirements

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different forms of requirements

- Consider finite sets of MSCs, given as an enumerated set.

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different forms of requirements

- Consider finite sets of MSCs, given as an enumerated set.
- Consider MSGs, that may describe an infinite set of MSCs.

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different forms of requirements

- Consider finite sets of MSCs, given as an enumerated set.
- Consider MSGs, that may describe an infinite set of MSCs.
- Consider MSCs whose set of linearisations is a regular word language.

# Problem variants (1)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different forms of requirements

- Consider finite sets of MSCs, given as an enumerated set.
- Consider MSGs, that may describe an infinite set of MSCs.
- Consider MSCs whose set of linearisations is a regular word language.
- Consider MSGs that are non-local choice.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

- Consider CFMs without synchronisation messages.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

- Consider CFMs without synchronisation messages.
- Allow CFMs that may deadlock. Possibly, a realisation deadlocks.
- Forbid CFMs that deadlock. No realisation will ever deadlock.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

- Consider CFMs without synchronisation messages.
- Allow CFMs that may deadlock. Possibly, a realisation deadlocks.
- Forbid CFMs that deadlock. No realisation will ever deadlock.
- Consider CFMs that are deterministic.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

- Consider CFMs without synchronisation messages.
- Allow CFMs that may deadlock. Possibly, a realisation deadlocks.
- Forbid CFMs that deadlock. No realisation will ever deadlock.
- Consider CFMs that are deterministic.
- Consider CFMs that are bounded.

# Problem variants (2)

## Realisability problem

INPUT: a set of MSCs

OUTPUT: a CFM $\mathcal{A}$ such that $\mathcal{L}(\mathcal{A})$ equals the set of input MSCs.

## Different system models

- Consider CFMs without synchronisation messages.
- Allow CFMs that may deadlock. Possibly, a realisation deadlocks.
- Forbid CFMs that deadlock. No realisation will ever deadlock.
- Consider CFMs that are deterministic.
- Consider CFMs that are bounded.
- ......

# Today's lecture

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

This is the setting of the previous lecture, but now focusing on deadlock-free CFMs

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

This is the setting of the previous lecture, but now focusing on deadlock-free CFMs

## Results:

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

This is the setting of the previous lecture, but now focusing on deadlock-free CFMs

## Results:

1. Conditions for realisability of a finite set of MSCs by a deadlock-free weak CFM.

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

This is the setting of the previous lecture, but now focusing on deadlock-free CFMs

## Results:

1. Conditions for realisability of a finite set of MSCs by a deadlock-free weak CFM.
2. Checking safe realisability by deadlock-free CFMs is in P.

# Today's lecture

## Today's setting

Realisation of a finite set of MSCs by a deadlock-free weak CFM.

Realisation of a finite set of well-formed words (= language) by a deadlock-free weak CFM.

This is known as safe realisability.

This is the setting of the previous lecture, but now focusing on deadlock-free CFMs

## Results:

1. Conditions for realisability of a finite set of MSCs by a deadlock-free weak CFM.
2. Checking safe realisability by deadlock-free CFMs is in P.

   (Realisability for weak CFMs that may deadlock is co-NP complete.)

# Safe realisability

Possibly a set of MSCs is realisable only by a CFM that may deadlock

# Safe realisability

Possibly a set of MSCs is realisable only by a CFM that may deadlock

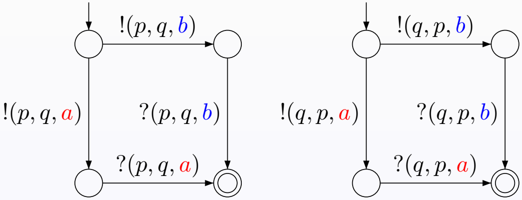

process $p$ and $q$ have to agree on either $a$ or $b$

# Safe realisability

Possibly a set of MSCs is realisable only by a CFM that may deadlock


process $p$ and $q$ have to agree on either $a$ or $b$

Realisation of $\{ M_1, M_2 \}$ by a weak CFM:



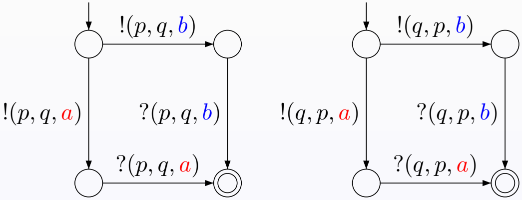

process $p$ process $q$

# Safe realisability

Possibly a set of MSCs is realisable only by a CFM that may deadlock


process $p$ and $q$ have to agree on either $a$ or $b$

Realisation of $\{ M_1, M_2 \}$ by a weak CFM:




process $p$ process $q$

Deadlock occurs when, e.g., $p$ sends $a$ and $q$ sends $b$

# Safe realisability

## Definition (Safe realisability)

1. MSC $M$ is safely realisable whenever $\{M\} = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.
2. A finite set $\{M_1, \ldots, M_n\}$ of MSCs is safely realisable whenever $\{M_1, \ldots, M_n\} = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.
3. MSG $G$ is safely realisable whenever $\mathcal{L}(G) = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.

# Safe realisability

## Definition (Safe realisability)

1. MSC $M$ is safely realisable whenever $\{M\} = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.
2. A finite set $\{M_1, \ldots, M_n\}$ of MSCs is safely realisable whenever $\{M_1, \ldots, M_n\} = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.
3. MSG $G$ is safely realisable whenever $\mathcal{L}(G) = \mathcal{L}(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.

## Phrased using linearisations

$L \subseteq Act^*$ is safely realisable if $L = Lin(\mathcal{A})$ for some deadlock-free CFM $\mathcal{A}$.

## Note:

Safe realisability implies realisability, but the converse does not hold.

# Overview

1. Safe realisability
2. Closure and inference revisited
3. Characterisation and complexity of safe realisability

# Weak closure

# Weak closure

**Definition (Inference relation and closure)**

For well-formed $L \subseteq Act^*$, and well-formed word $w \in Act^*$, let:

$$L \models w \quad \text{iff} \quad (\forall p \in \mathcal{P}.\, \exists v \in L.\, w \restriction p = v \restriction p)$$

Language $L$ is closed under $\models$ whenever for every $w \in Act^*$, it holds: $L \models w$ implies $w \in L$.

# Weak closure

## Definition (Inference relation and closure)

For well-formed $L \subseteq Act^*$, and well-formed word $w \in Act^*$, let:

$$L \models w \quad \text{iff} \quad (\forall p \in \mathcal{P}. \exists v \in L. w \upharpoonright p = v \upharpoonright p)$$

Language $L$ is closed under $\models$ whenever for every $w \in Act^*$, it holds: $L \models w$ implies $w \in L$.

## Definition (Weak closure)

Language $L$ is weakly closed under $\models$ whenever for every well-formed prefix $w$ of some word in $L$, it holds $L \models w$ implies $w \in L$.

Weak closure thus restricts closure under $\models$ to well-formed prefixes in $L$ only. So far, closure was required for all $w \in Act^*$.

# Deadlock-free closure

For language $L$, let $\mathit{pref}(L) = \{ w \mid \exists u.\, w \cdot u \in L \}$ the set of prefixes of $L$.

# Deadlock-free closure

For language $L$, let $\mathit{pref}(L) = \{w \mid \exists u.\, w{\cdot}u \in L\}$ the set of prefixes of $L$.

## Definition ((Deadlock-free) Inference relation)

For well-formed $L \subseteq \mathit{Act}^*$, and proper word $w \in \mathit{Act}^*$, i.e., $w$ is a prefix of a well-formed word, let:

$$L \models^{\mathit{df}} w \quad \text{iff} \quad (\forall p \in \mathcal{P}.\, \exists v \in \mathit{pref}(L).\, w \upharpoonright p \text{ is a prefix of } v \upharpoonright p)$$

# Deadlock-free closure

For language $L$, let $pref(L) = \{w \mid \exists u.\, w{\cdot}u \in L\}$ the set of prefixes of $L$.

## Definition ((Deadlock-free) Inference relation)

For well-formed $L \subseteq Act^*$, and proper word $w \in Act^*$, i.e., $w$ is a prefix of a well-formed word, let:

$$L \models^{df} w \quad \text{iff} \quad (\forall p \in \mathcal{P}.\, \exists v \in pref(L).\, w \upharpoonright p \text{ is a prefix of } v \upharpoonright p)$$

## Definition (Closure under $\models^{df}$)

Language $L$ is closed under $\models^{df}$ whenever $L \models^{df} w$ implies $w \in pref(L)$.

# Deadlock-free closure

For language $L$, let $pref(L) = \{w \mid \exists u.\, w \cdot u \in L\}$ the set of prefixes of $L$.

## Definition ((Deadlock-free) Inference relation)

For well-formed $L \subseteq Act^*$, and proper word $w \in Act^*$, i.e., $w$ is a prefix of a well-formed word, let:

$$L \models^{df} w \quad \text{iff} \quad (\forall p \in \mathcal{P}.\, \exists v \in pref(L).\, w \upharpoonright p \text{ is a prefix of } v \upharpoonright p)$$

## Definition (Closure under $\models^{df}$)

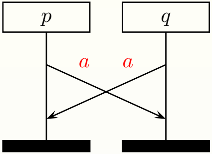
Language $L$ is closed under $\models^{df}$ whenever $L \models^{df} w$ implies $w \in pref(L)$.

## Intuition

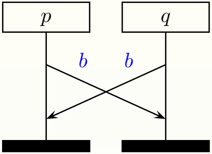
The closure condition asserts that the set of partial MSCs (i.e., prefixes of $L$) can be constructed from the projections of the MSCs in $L$ onto individual processes.

# Example

MSC $M_1$:

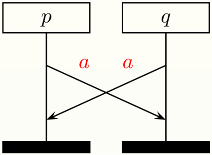



MSC $M_2$:

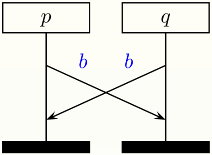


# Example

MSC $M_1$:




MSC $M_2$:




## Example

$L = Lin(\{M_1, M_2\})$ is not closed under $\models^{df}$:

$$w \ = !(p, q, a)!(q, p, b) \notin \ pref(L)$$

But: $L \models^{df} w$ since $w$ is a proper prefix of a well-formed word, and

- for process $p$, there exists $u \in L$ with $w \upharpoonright p = !(p, q, a) \in pref(\{u \upharpoonright p\})$, and
- for process $q$, there exists $v \in L$ with $w \upharpoonright q = !(q, p, b) \in pref(\{v \upharpoonright q\})$.

Note that $L$ is closed under $\models$. So this shows that closure under $\models$ does not imply closure under $\models^{df}$.

# Deadlock-free weak CFM are closed under $\models^{df}$

**Lemma:**

For every deadlock-free weak CFM $\mathcal{A}$, $Lin(\mathcal{A})$ is closed under $\models^{df}$.

**Proof.**

Similar proof strategy as for the closure of weak CFMs under $\models$ (see previous lecture).

# Deadlock-free weak CFM are closed under $\models^{df}$

## Lemma:

For every deadlock-free weak CFM $\mathcal{A}$, $Lin(\mathcal{A})$ is closed under $\models^{df}$.

## Proof.

Similar proof strategy as for the closure of weak CFMs under $\models$ (see previous lecture). Basic intuition is that if $w \upharpoonright p$ is a prefix of $v^p \upharpoonright p$, then from the point of view of process $p$, $w$ can be prolonged with a word $u$, say, such that $w \cdot u = v^p$.

# Deadlock-free weak CFM are closed under $\models^{df}$

**Lemma:**

For every deadlock-free weak CFM $\mathcal{A}$, $Lin(\mathcal{A})$ is closed under $\models^{df}$.

**Proof.**

Similar proof strategy as for the closure of weak CFMs under $\models$ (see previous lecture). Basic intuition is that if $w \upharpoonright p$ is a prefix of $v^p \upharpoonright p$, then from the point of view of process $p$, $w$ can be prolonged with a word $u$, say, such that $w \cdot u = v^p$. This applies to all processes, and as the weak CFM is deadlock-free, such continuation is always possible. $\square$

# Overview

1. Safe realisability
2. Closure and inference revisited
3. Characterisation and complexity of safe realisability

# Characterisation of safe realisability

**Theorem:** [Alur *et al.*, 2001]

$L \subseteq Act^*$ is safely realisable iff $L$ is weakly closed under $\models$ and closed under $\models^{df}$.

# Characterisation of safe realisability

**Theorem:** [Alur *et al.*, 2001]

$L \subseteq Act^*$ is safely realisable iff $L$ is weakly closed under $\models$ and closed under $\models^{df}$.

**Proof**

On the black board.

# Characterisation of safe realisability

**Theorem:** [Alur *et al.*, 2001]

$L \subseteq Act^*$ is safely realisable iff $L$ is weakly closed under $\models$ and closed under $\models^{df}$.

**Proof**

On the black board.

**Corollary**

The finite set of MSCs $\{M_1, \ldots, M_n\}$ is safely realisable iff $\bigcup_{i=1}^{n} Lin(M_i)$ is closed under $\models$ and $\models^{df}$.

# Characterisation of safe realisability

**Theorem**

For any well-formed $L \subseteq Act^*$:

$$L \text{ is regular and closed under } \models$$
$$\text{if and only if}$$
$$L = Lin(\mathcal{A}) \text{ for some } \forall\text{-bounded weak CFM } \mathcal{A}.$$

# Characterisation of safe realisability

**Theorem**

For any well-formed $L \subseteq Act^*$:

$$L \text{ is regular and closed under } \models$$
$$\text{if and only if}$$
$$L = Lin(\mathcal{A}) \text{ for some } \forall\text{-bounded weak CFM } \mathcal{A}.$$

**Theorem**

For any well-formed $L \subseteq Act^*$:

$$L \text{ is regular, weakly closed under } \models \text{ and closed under } \models^{df}$$
$$\text{if and only if}$$
$$L = Lin(\mathcal{A}) \text{ for some } \forall\text{-bounded deadlock-free weak CFM } \mathcal{A}.$$

# Complexity of safe realisability

**Theorem:** [Alur *et al.*, 2001]

The decision problem “is a given set of MSCs safely realisable?” is in P.

# Complexity of safe realisability

**Theorem:** [Alur *et al.*, 2001]

The decision problem "is a given set of MSCs safely realisable?" is in P.

**Proof**

1. For a given finite set of MSCs, safe realisability can be checked in time $\mathcal{O}((n^2 + r) \cdot k)$ where $k$ is the number of processes, $n$ the number of MSCs, and $r$ the number of events in all MSCs together.
2. If the MSCs are not safely realisable, the algorithm returns an MSC which is implied, but not included in the input set of MSCs.

(We skip the details in this lecture.)