Theoretical Foundations of the UML

Lecture 10: Realisability → Complexity

Joost-Pieter Katoen

Lehrstuhl für Informatik 2
Software Modeling and Verification Group

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

May 19, 2020

Realisability problem

input: a finite set of MSCs $\{M_1, \dots, M_n\}$

output: a weak CFM A that realises $\{M_1, \dots, M_n\}$

no sync message

acceptance condition

$L(A) = \{M_1, \dots, M_n\}$

$$F = \prod_{p \in P} F_p$$

Main theorem of Lecture 9

Finite $L \subseteq Act^*$ is realisable (by a weak CFM)

if and only if

L is closed under $\models$

inference relation

Recall:

linearisations of $\{M_1, \dots, M_n\}$

1. well-formed $L \subseteq Act^*$, $w \in Act^*$ is well-formed.

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

2. L is closed under $\models$ iff ($L \models w$ implies $w \in L$)

Topic of today: $\underbrace{\text{how hard}}_{\text{what is the complexity}}$ is it of checking

whether $L \subseteq Act^*$ is closed under $\models$?

Result: this problem is co-NP complete.

Corollary: checking whether a finite set of MSCs is realisable by a weak CFM is coNP-complete.

Explanation

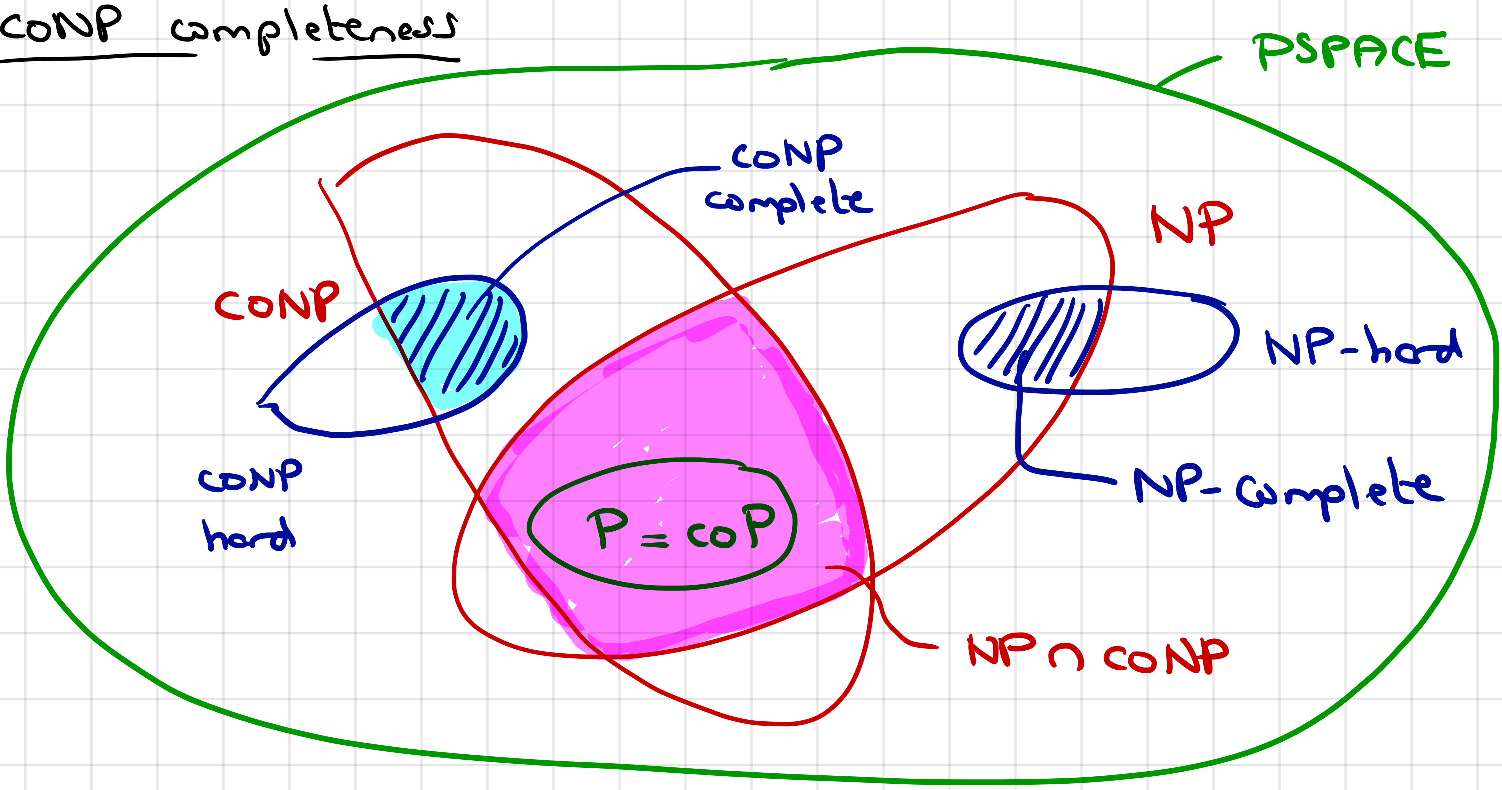
1. What is coNP completeness?

2. Join dependency problem (JDP)

3. Polynomial reduction of the JDP onto the realisability problem (is L closed under $\models$?)

   The realisability problem lies in coNP.

coNP completeness



PSPACE $\subseteq$ EXPTIME

Decision problem $H \in P$, then $\overline{H} \in coP \longrightarrow P = coP$

It is believed that $coNP \neq NP$, $(P \neq NP)$

Examples:

- is a given number a prime number? P
- SAT-problem, Boolean formula NP-complete

  $(x_1 \wedge x_2) \vee (x_1 \wedge \overline{x_2} \wedge x_3)$ etc.
- determine whether a Boolean formula is a tautology? coNP-complete

Simple characterisation of coNP:

the class of problems for which <u>efficiently</u> verifiable proofs of <u>counterexamples</u> exist

<u>Alternatively</u>: coNP is the class of all decision problems H such that $\overline{H} \in NP$.

2 To show that our decision problem:

is $L \subseteq Act^*$ closed under $\models$? (*)

is coNP-complete, we <u>identify</u> a <u>decision problem</u> that is coNP-hard and provide a polynomial reduction to (*).

Join Dependency Problem (JDP)

JDP example:

inputs: 1. universe $\mathcal{U} = \{a, b, c\}$

2. cardinality $k \in \mathbb{N}$, e.g. $k = 4$

3. relation $R \subseteq \mathcal{U}^k$, e.g.

records in a database

$$R = \{(a,a,a,b), (a,a,b,a), (a,b,a,a), (b,a,a,a)\}$$

4. index set Ind over $[1..k]$, e.g.

$$I = \{\underbrace{\{1,2,3\}}_{I_1}, \{2,3,4\}, \underbrace{\{1,3,4\}}_{I_3}\}$$

subtables

$R \upharpoonright I_3$

$$\longrightarrow \begin{bmatrix} a & a & a & b \\ a & a & b & a \\ a & b & a & a \\ b & a & a & a \end{bmatrix}$$

$R \upharpoonright I_1$

JDP: $\forall \bar{a} \in \mathcal{U}^4.\ (\forall I.\ \bar{a} \upharpoonright I \in R \upharpoonright I)$ implies $\bar{a} \in R$?

can we reconstruct the database R from the subtables $R \upharpoonright I_1, \ldots, R \upharpoonright I_m$?

for our example: (*) $\left(\forall I.\ \bar{a}\lceil I \in R\lceil I\right) \rightsquigarrow \bar{a} \in R$

a) $\bar{a} = (b,b,b,b)$ e.g. $\bar{a}\lceil I_1 = \bar{a}\lceil\{1,2,3\} = (b,b,b)$

but $(b,b,b) \notin R\lceil I$, so no obligation

for $\bar{a}$ to be in R. Indeed $\bar{a} \notin R$.

b) $\bar{a} = (a,a,a,a) \notin R$.

b1) $\bar{a}\lceil I_1 = (a,a,a) \in R\lceil I_1$ ✓

b2) $\bar{a}\lceil I_2 = (a,a,a) \in R\lceil I_2$ ✓

b3) $\bar{a}\lceil I_3 = (a,a,a) \in R\lceil I_3$ ✓

not a join dependency

<u>Intuition</u>: by combining the subtables $R\lceil I_1$, $R\lceil I_2$, $R\lceil I_3$ would imply that $(a,a,a,a) \in R$, but $(a,a,a,a) \notin R$

Definition (Join dependency problem)

let $\mathcal{U}$ be a finite set of elements (= universe)

$k \in \mathbb{N}$ (cardinality)

database records $R \subseteq \mathcal{U}^k$ $\quad \bar{a} = (a_1, \dots, a_k),\ a_i \in \mathcal{U}$

index sets $Ind = \{I_1, \dots, I_m\} \subseteq [1..k]$

$$I_j = \{i_1, \dots, i_{m_j}\} \qquad \bar{a} \restriction I_j = (a_{i_1}, \dots, a_{i_{m_j}})$$

$$R \restriction I_j = \{\bar{b} \in \mathcal{U}^m \mid \exists \bar{a} \in R.\ \bar{a} \restriction I_j = \bar{b}\}$$

Constraint of Ind: every $i \in [1..k]$ appears at least once in some I_j

JDP: does there exist a join dependency:

for all $\bar{a} \in \mathcal{U}^k$, it holds

$$(\forall I_j \in Ind.\ \bar{a} \restriction I_j \in R \restriction I_j) \text{ implies } \bar{a} \in R.$$

Intuition: relation R (= database) can be reconstructed by joining multiple tables each having a subset of the attributes in the records stored by R.

Theorem [Maier, Sagiv, Yannakakis, 1981]

JDP is coNP-complete

Theorem The decision problem "is a given finite set of MSCs realisable (by a weak CFM) coNP-complete"

Proof

(1) This decision problem lies in coNP.

(2) This decision problem is coNP-hard.

---

(1) **Lemma** The decision problem realisability by a weak CFM is in coNP.

Proof (sketch) Show that the complement of the realisability problem lies in NP.

To check that $\{M_1, \dots, M_n\}$ is <u>not</u> realisable is in NP we pursue as follows:

a. Guess nondeterministically for every process $p \in P$ an MSC $M_p \in \{M_1, \dots, M_n\}$. Let w_p be $M_p \upharpoonright_p$ is the sequence of actions occurring at process p in M_p.

b. Check that the projections w_{p_i} (for every process $p_i \in P$) are <u>consistent</u>, i.e., their combination is a well-formed complete MSC M.

c. Check whether $M \notin \{M_1, \dots, M_n\}$.

Ergo: we can check nonrealisability in NP. $\boxtimes$

2 <u>Lemma</u>: The realisability problem is coNP-hard.

<u>Proof</u>: provide a polynomial reduction from the JDP onto the realisability problem:

$$\underbrace{(U, k, R \subseteq U^k, Ind)}_{\text{instance of JDP}} \longmapsto \underbrace{\{M_1, \dots, M_{|R|}\}}_{\text{instance of the realisability problem}}$$

set of MSCs

$|R|$: # tuples in R

such that: $(U, k, R, Ind) \in JDP$

iff $\{M_1, \dots, M_{|R|}\}$ is realisable (by a weak CFM)

iff $\{M_1, \dots, M_{|R|}\}$ is closed under $\models$

Polynomial reduction:

- as Ind may contain several index sets multiple times we assume w.l.o.g. that every $i \in [1..k]$ belongs to at least two sets $I_j, I_{j'} \in Ind$. (If this is not the case, just duplicate I_j in Ind.)

- $\underbrace{Ind = \{I_1, \dots, I_m\}}_{\text{part JDP}} \mapsto \underbrace{P = \{p_1, \dots, p_m\}}_{\text{realisability}}$

  i.e. one process for each index set

- $R = \{\bar{a}_1, \dots, \bar{a}_n\}$ with $\bar{a}_i \in U^k$

  $\mapsto$ MSCs $\{M_{\bar{a}_1}, \dots, M_{\bar{a}_n}\}$

  every MSC $M_{\bar{a}_j}$ has the same structure, i.e., the same message exchanges; only the message content differs.

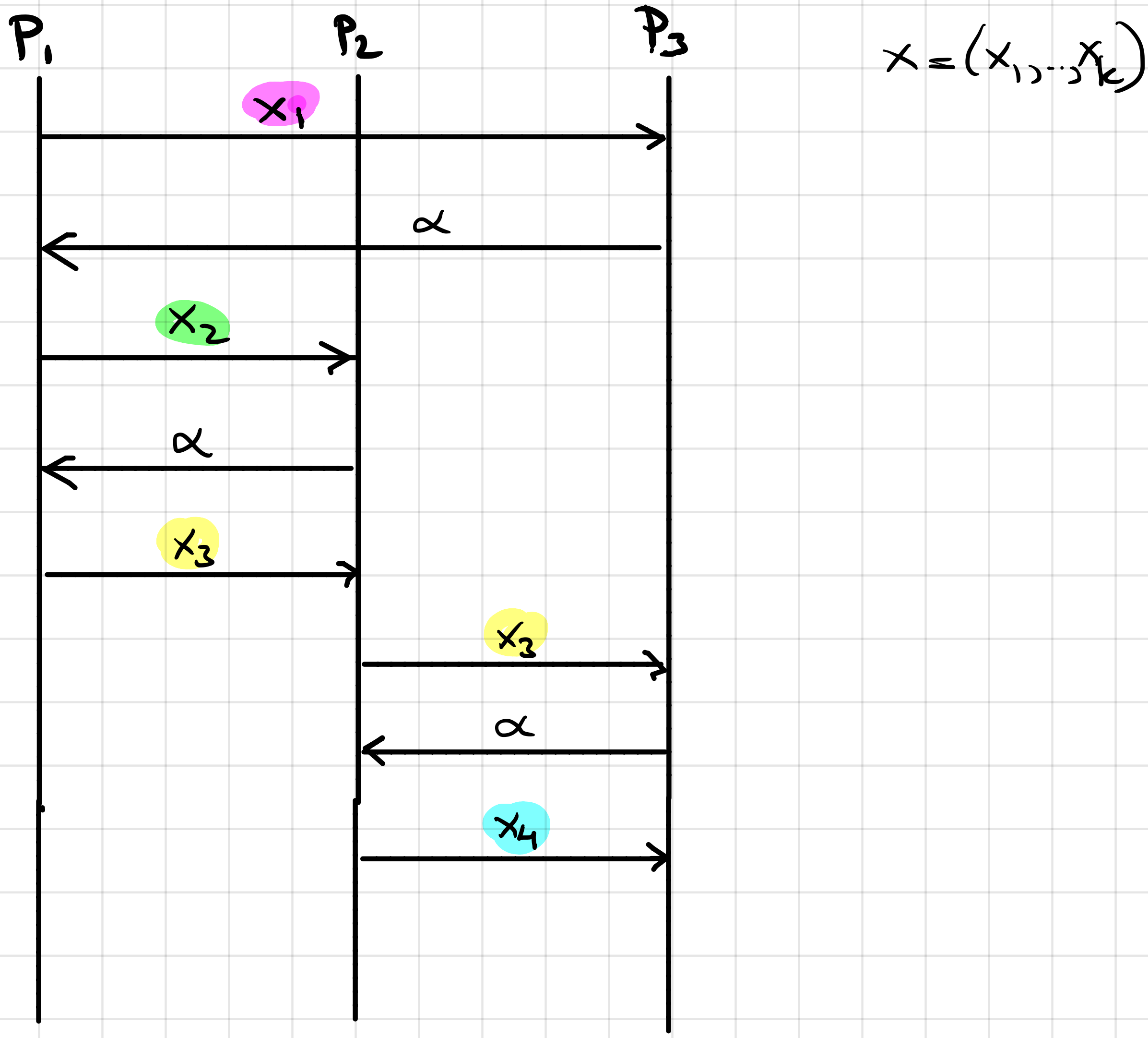
  So, for every record in database R, we have 1 MSC.

These MSCs are defined as follows.

By example: $\mathrm{Ind} = \{I_1, I_2, I_3\}$

$I_1 = \{1, 2, 3\}$ $\quad$ $I_2 = \{2, 3, 4\}$ $\quad$ $I_3 = \{1, 3, 4\}$

Structure of MSC $M_{\overline{x}}$ for $\overline{x} \in \{\overline{a}_1, \dots, \overline{a}_n\}$:



For tuple $\overline{a} = (a_1, \dots, a_k) \in R$ we obtain $M_{\overline{a}}$ by replacing $\overline{x}$ by $\overline{a}$. The finite set of MSCs

$$\mathcal{M} = \{ M_{\overline{a}} \mid \overline{a} \in R \}$$

Clearly, this reduction can be done in polynomial time

Remarks: $M_{\bar{x}} \restriction p$ contains (either sends or receives) message x_j if and only if $j \in I_p$

In addition, MSC $M_{\bar{x}}$ has a unique linearisation, i.e. $\mathrm{Lin}(M_{\bar{x}})$ is a singleton set.

$$\boxed{\begin{array}{l}\text{Claim: } (\mathcal{U}, k, R, \mathrm{Ind}) \text{ is a join dependency} \\ \qquad\qquad \text{if and only if} \\ \qquad \{M_1, \ldots, M_{|R|}\} \text{ is a realisable} \\ \qquad\qquad\qquad \text{(by a weak CFM)}\end{array}}$$

Proof: "$\Leftarrow$" By contraposition. Assume that $\{M_1, \ldots, M_{|R|}\}$ is realisable and $(\mathcal{U}, k, R, \mathrm{Ind})$ is not a join dependency. Then there exists $\bar{a} = (a_1, \ldots, a_k) \in \mathcal{U}^k$ such that

$$\bar{a} \restriction I \in R \restriction I \text{ for all } I \in \mathrm{Ind} \qquad (*)$$
$$\text{but } \bar{a} \notin R.$$

Take $I_j \in \mathrm{Ind}$. Since $\bar{a} \restriction I_j \in R \restriction I_j$ there is a $\bar{b}^j \in R$ such that $\bar{a} \restriction I_j = \bar{b}^j \restriction I_j$ ~~$\in R$~~.

Consider the MSC $M_{\bar{a}}$. By construction, $M_{\bar{a}}\restriction j$ only "depends" on $\bar{a}\restriction I_j$. Hence, since $\bar{a}\restriction I_j = \bar{b}^j \restriction I_j$, it follows $\boxed{M_{\bar{a}}\restriction j = M_{\bar{b}^j}\restriction j.}$

This applies to all $I_j \in Ind$, thus $M_{\bar{b}^j} \in \mathcal{M} = \{M_1, \dots, M_{|R|}\}$. This applies to any j so

$$M_{\bar{b}^1}, \dots M_{\bar{b}^m} \text{ all belong to } \mathcal{M}.$$

$\underbrace{\text{Since } \{M_1, \dots, M_{|R|}\} \text{ is realisable}}_{\{M_1, \dots, M_{|R|}\} \text{ is closed under } \models}$, $M_{\bar{a}} \in \mathcal{M}$.

Contradiction to $\bar{a} \notin R$.

"$\Longrightarrow$": goes along similar lines. Let $(\mathcal{U}, k, R, Ind)$ be a join dependency. By contraposition, assume $\underbrace{\{M_1, \dots, M_{|R|}\}}_{\mathcal{M}}$ is <u>not</u> realisable. But if $\mathcal{M} \vdash M \notin \mathcal{M}$, we can "read off" from $\{M_1, \dots, M_{|R|}\}$ tuples $\bar{b}^j \in R\restriction I_j$ for each j such that there is a k-tuple $\bar{a} \in \mathcal{U}^k$ such that $\bar{a}\restriction I_j = \bar{b}^j$ for each j, but $\bar{a} \notin R$. Contradiction

$\boxtimes$.