



Definitions and proofs
of week 2 revisited

Définition 2.3.3 (Exécution causalement ordonnée [Charron-Bost et al., 1996]). Soit un MSC $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$, μ admet une exécution causalement ordonnée si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $\mathbf{m}' = \{s', r'\}$:

$$(\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}')) \wedge (s \prec s') \implies r \prec r' \quad (2.2)$$



Causally ordered MSCs An MSC is causally ordered if all the messages sent to the same process are received in an order which is consistent with the causal ordering of the corresponding send events. More formally, for an MSC $M = (\mathcal{E}, \rightarrow, \triangleleft, \lambda)$ we define an additional binary relation $\blacktriangleleft_M \subseteq \mathcal{E} \times \mathcal{E}$ that represents a constraint under the causal ordering semantics. In particular, given two receive events f_1 and f_2 , we have that $f_1 \blacktriangleleft_M f_2$ if both the following hold:

- $\lambda(f_1) \in \text{Rec}(_, q, _)$, $\lambda(f_2) \in \text{Rec}(_, q, _)$
- $e_1 \triangleleft f_1$ and $e_2 \triangleleft f_2$ for some $e_1, e_2 \in \mathcal{E}$, such that $e_1 \leq_M e_2$.

We let $\leq_M = (\rightarrow \cup \triangleleft \cup \blacktriangleleft_M)^*$. Note that $\leq_M \subseteq \blacktriangleleft_M$. We call $M \in \text{MSC}$ a *causally ordered (CO) MSC* if \leq_M is a partial order. The set of causally ordered MSCs $M \in \text{MSC}$ is denoted by MSC_{co} .

Proposition 2.1. The set MSC_{co} of causally ordered MSCs is MSO-definable.

Proof. The set of causally ordered MSCs can be defined using the MSO formula

$$\varphi_{\text{co}} = \neg \exists x. \exists y. \left(\bigvee_{\substack{q \in \mathbb{P} \\ a, b \in \text{Send}(_, q, _)}} \lambda(x) = a \wedge \lambda(y) = b \wedge x \leq y \wedge \exists x'. \exists y'. \left(\begin{array}{c} x \triangleleft x' \\ y \triangleleft y' \\ y' \rightarrow^+ x' \end{array} \wedge \right) \right)$$

The property φ_{co} says that there cannot be two send events x and y , with the same recipient, such that $x \leq y$ and their corresponding receive events x' and y' happen in the opposite order, i.e. $y' \rightarrow^+ x'$. The set MSC_{co} of causally ordered MSCs is therefore MSO-definable as $\text{MSC}_{\text{co}} = L(\varphi_{\text{co}})$. \square

Knowing that MSC_{co} is MSO-definable, Theorem 1.5 can be restated for $\text{com} = \text{co}$.

Theorem 2.3. *The bounded model-checking problem for $\text{com} = \text{co}$ is decidable.*

Proof. By Proposition 2.1, $\text{MSC}_{\text{co}} = L(\varphi_{\text{co}})$. Given a system \mathcal{S} , we have that $L_{\text{co}}(\mathcal{S}) = L_{\text{p2p}}(\mathcal{S}) \cap L(\varphi_{\text{co}})$. Therefore, we can rewrite the bounded model checking problem for $\text{com} = \text{co}$ as

$$\begin{aligned} L_{\text{co}}(\mathcal{S}) \cap \text{MSC}^{k\text{-stw}} &\subseteq L(\varphi) \\ \iff L_{\text{p2p}}(\mathcal{S}) \cap L(\varphi_{\text{co}}) \cap \text{MSC}^{k\text{-stw}} &\subseteq L(\varphi) \\ \iff L_{\text{p2p}}(\mathcal{S}) \cap \text{MSC}^{k\text{-stw}} &\subseteq L(\varphi) \cup L(\neg\varphi_{\text{co}}) \\ \iff L_{\text{p2p}}(\mathcal{S}) \cap \text{MSC}^{k\text{-stw}} &\subseteq L(\varphi \vee \neg\varphi_{\text{co}}). \end{aligned}$$

The latter is decidable due to Fact 1.4. □

Decidability of bounded
model checking for p2p



New stuff

Lemma 2.1. *Every prefix of a causally ordered MSC is a causally ordered MSC.*

Proof. Let $M = (\mathcal{E}, \rightarrow, \triangleleft, \lambda) \in \text{MSC}_{\text{co}}$ and let $M_0 = (\mathcal{E}_0, \rightarrow_0, \triangleleft_0, \lambda_0)$ be a prefix of M . By contradiction, suppose that M_0 is not a causally ordered MSC. Then, there are distinct $e, f \in \mathcal{E}_0$ such that $e \triangleleft_{M_0} f$ and $f \triangleleft_{M_0} e$, with $\triangleleft_{M_0} = (\rightarrow_0 \cup \triangleleft_0 \cup \blacktriangleleft_{M_0})^*$. As $\mathcal{E}_0 \subseteq \mathcal{E}$, we have that $\rightarrow_0 \subseteq \rightarrow$, $\triangleleft_0 \subseteq \triangleleft$, and $\blacktriangleleft_{M_0} \subseteq \blacktriangleleft_M$. Finally, we have that $\triangleleft_{M_0} \subseteq \triangleleft_M$ and M cannot be a causally ordered MSC, which is a contradiction. \square

Lemma 1.2 can be easily extended to $\text{com} = \text{co}$.

Lemma 2.2. *For all $\text{com} \in \{\text{p2p}, \text{mb}, \text{co}\}$, $L_{\text{com}}(\mathcal{S})$ is prefix-closed: $\text{Pref}(L_{\text{com}}(\mathcal{S})) \subseteq L_{\text{com}}(\mathcal{S})$.*

Proof. Follows from Lemma 2.1. \square

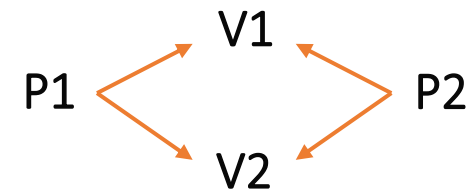
Définition 2.3.3 (Exécution causalement ordonnée [Charron-Bost et al., 1996]). Soit un MSC $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$, μ admet une exécution causalement ordonnée si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $\mathbf{m}' = \{s', r'\}$:


$$(\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}')) \wedge (s \prec s') \implies r \prec r' \quad (2.2)$$

Theorem 2.6. *The following problem is undecidable: Given finite sets \mathbb{P} and \mathbb{M} as well as a communicating system \mathcal{S} , is every MSC in $L_{co}(\mathcal{S})$ weakly synchronous?*

Proof. The proof is essentially identical to the p2p case. We do the same reduction from the Post correspondence problem. Recall from the proof of Theorem 1.10 that we consider a system \mathcal{S} with four machines (P1, P2, V1, V2), where we have unidirectional communication channels from provers to verifiers. In particular notice that all the possible behaviours of \mathcal{S} are causally ordered, i.e. $L_{p2p}(\mathcal{S}) \subseteq \text{MSC}_{co}$; according to how we built our system \mathcal{S} , it is impossible to have a pair of causally-related send events of P1 and P2¹, hence the causal ordering binary relation \blacktriangleleft_M will be empty for any $M \in L_{p2p}(\mathcal{S})$ (i.e. causal ordering is already ensured by any possible p2p behaviour of \mathcal{S}). The rest of the proof is identical to the p2p case. \square

¹Notice that there is no channel between P1 and P2, and we only have unidirectional communication channels from provers to verifiers.





Proposition 2.3. The set of weakly synchronous causally ordered MSCs has unbounded special tree-width.

Proof. Suppose that the set of weakly synchronous causally ordered MSCs is STW-bounded. By Proposition 2.2 and Theorem 2.5, we have that the synchronicity problem for the class of weakly synchronous causally ordered MSCs would be decidable. This is a contradiction, since Theorem 2.6 states that this problem is undecidable. \square




Table 2: Summary of the decidability of the synchronizability problem in various classes

	P2P	CAUSAL ORDERING	MAILBOX
Weakly synchronous	Undecidable [Thm. 1.10]	Undecidable [Thm. 2.6]	EXPTIME [Thm. 1.9]
Weakly k -synchronous	Decidable [Thm. 2.7]		



Doubts

Définition 2.2.5 (Réalisation en boîte aux lettres). Soit $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$ un MSC. On dit alors que μ est *mb-réalisable* s'il existe une linéarisation $e = a_1 \cdots a_n$ avec un ordre total $<$ telle que, pour toute paire d'évènements $i < j$ telle que $a_i = s(p, q, m)$ et $a_j = s(p', q, m')$, soit a_j est non couplé, soit il existe i', j' tel que $a_i \vdash a_{i'}$, $a_j \vdash a_{j'}$ et $i' < j'$.

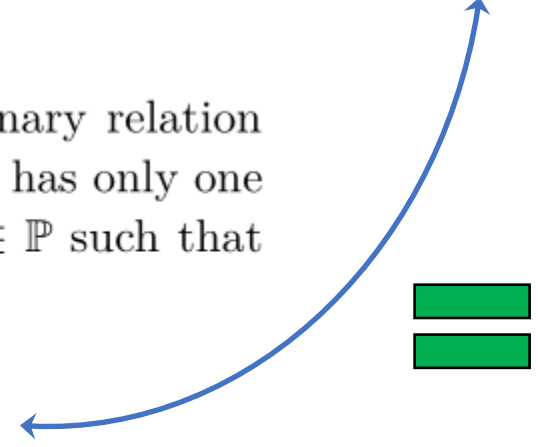
Mailbox MSCs. For an MSC $M = (\mathcal{E}, \rightarrow, \triangleleft, \lambda)$, we define an additional binary relation that represents a constraint under the mailbox semantics, where each process has only one incoming channel. Let $\sqsubset_M \subseteq \mathcal{E} \times \mathcal{E}$ be defined by: $e_1 \sqsubset_M e_2$ if there is $q \in \mathbb{P}$ such that $\lambda(e_1) \in Send(_, q, _)$, $\lambda(e_2) \in Send(_, q, _)$, and one of the following holds:

- $e_1 \in Matched(M)$ and $e_2 \in Unm(M)$, or
- $e_1 \triangleleft f_1$ and $e_2 \triangleleft f_2$ for some $f_1, f_2 \in \mathcal{E}_q$ such that $f_1 \rightarrow^+ f_2$.

We let $\preceq_M = (\rightarrow \cup \triangleleft \cup \sqsubset_M)^*$. Note that $\leq_M \subseteq \preceq_M$. We call $M \in \text{MSC}$ a *mailbox MSC* if \preceq_M is a partial order. Intuitively, this means that events can be scheduled in a way that corresponds to the mailbox semantics, i.e., with one incoming channel per process. Following the terminology in [8], we also say that a mailbox MSC satisfies *causal delivery*. The set of mailbox MSCs $M \in \text{MSC}$ is denoted by MSC_{mb} .

$\leq_M \subseteq \rightsquigarrow$. Similarly, a *mailbox linearization* of M is a total order $\rightsquigarrow \subseteq \mathcal{E} \times \mathcal{E}$ such that

That is, every mailbox linearization is a p2p linearization, but the converse is not necessarily true (Example 2). Note that an MSC is a mailbox MSC iff it has at least one mailbox linearization.



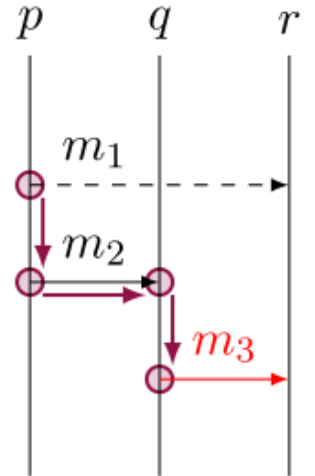
Définition 2.3.3 (Exécution causalement ordonnée [Charron-Bost et al., 1996]). Soit un MSC $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$, μ admet une exécution causalement ordonnée si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $\mathbf{m}' = \{s', r'\}$:

$$(\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}')) \wedge (s \prec s') \implies r \prec r' \quad (2.2)$$

Définition 2.3.6 (Livraison causale [Bouajjani et al., 2018a]). Soit $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$ un MSC, μ vérifie la livraison causale si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $s' \in \mathbf{m}'$:

$$(s \prec s') \wedge (\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}')) \implies (\mathbf{m}' = \{s'\}) \vee (\mathbf{m}' = \{s', r'\} \wedge (r' \not\prec r)) \quad (2.4)$$

Definition 4 (Causal Delivery). Let (Ev, λ, \prec) be an MSC. We say that it satisfies causal delivery if the MSC has a linearisation $e = a_1 \dots a_n$ such that for any two events $i \prec j$ such that $a_i = \text{send}(p, q, \mathbf{v})$ and $a_j = \text{send}(p', q, \mathbf{v}')$, either a_j is unmatched, or there are i', j' such that $a_i \vdash a_{i'}$, $a_j \vdash a_{j'}$, and $i' \prec j'$.



!1 !2 ?2 !3 ?3

Verifies def 2.3.6, but not def 4

(a)

Définition 2.2.5 (Réalisation en boîte aux lettres). Soit $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$ un MSC. On dit alors que μ est *mb-réalisable* s'il existe une linéarisation $e = a_1 \cdots a_n$ avec un ordre total $<$ telle que, pour toute paire d'évènements $i < j$ telle que $a_i = s(p, q, m)$ et $a_j = s(p', q, m')$, soit a_j est non couplé, soit il existe i', j' tel que $a_i \vdash a_{i'}$, $a_j \vdash a_{j'}$ et $i' < j'$.

Définition 2.3.6 (Livraison causale [Bouajjani et al., 2018a]). Soit $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$ un MSC, μ vérifie la livraison causale si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $s' \in \mathbf{m}'$:

$$(s \prec s') \wedge (\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}')) \implies (\mathbf{m}' = \{s'\}) \vee (\mathbf{m}' = \{s', r'\} \wedge (r' \not\prec r)) \quad (2.4)$$

Mailbox \rightarrow Causal delivery
Causal delivery \nrightarrow Mailbox

Définition 2.3.5 (Exécution $n-1$). Pour une exécution avec un ordre total $<$ sur ses actions, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $\mathbf{m}' = \{s', r'\}$:

$$\left(\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}') \right) \wedge (s < s') \implies r < r' \quad (2.3)$$

Mailbox without taking into consideration unmatched messages

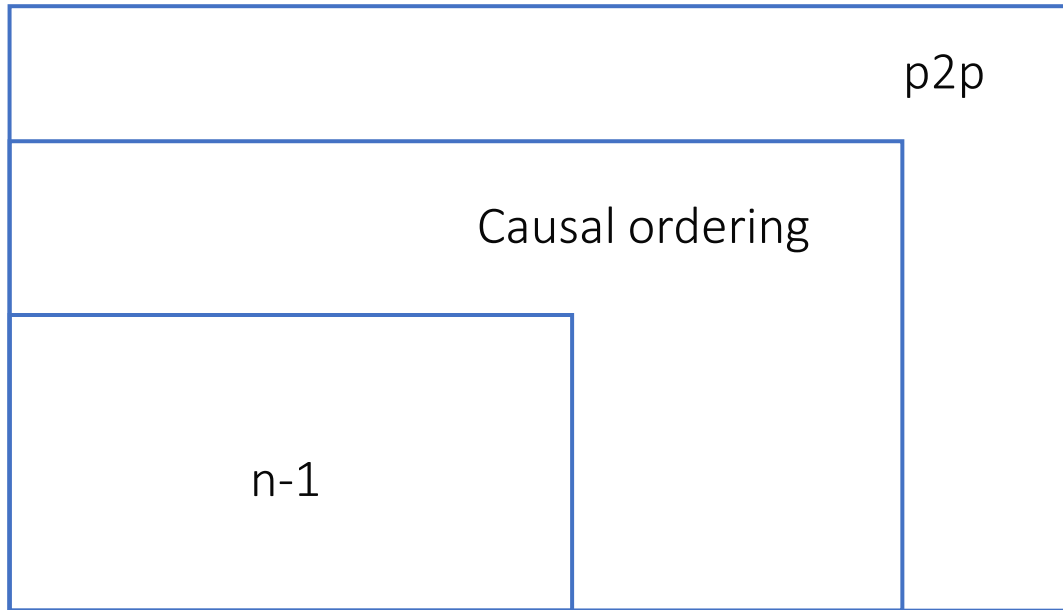
Définition 2.3.3 (Exécution causalement ordonnée [Charron-Bost et al., 1996]). Soit un MSC $\mu = (Ev, \lambda, \prec_{po}, \prec_{src})$, μ admet une exécution causalement ordonnée si, pour deux messages $m, m' \in \mathbb{V}$, tels que $\mathbf{m} = \{s, r\}$ et $\mathbf{m}' = \{s', r'\}$:

$$\left(\text{proc}_R(\mathbf{m}) = \text{proc}_R(\mathbf{m}') \right) \wedge (s \prec s') \implies r \prec r' \quad (2.2)$$

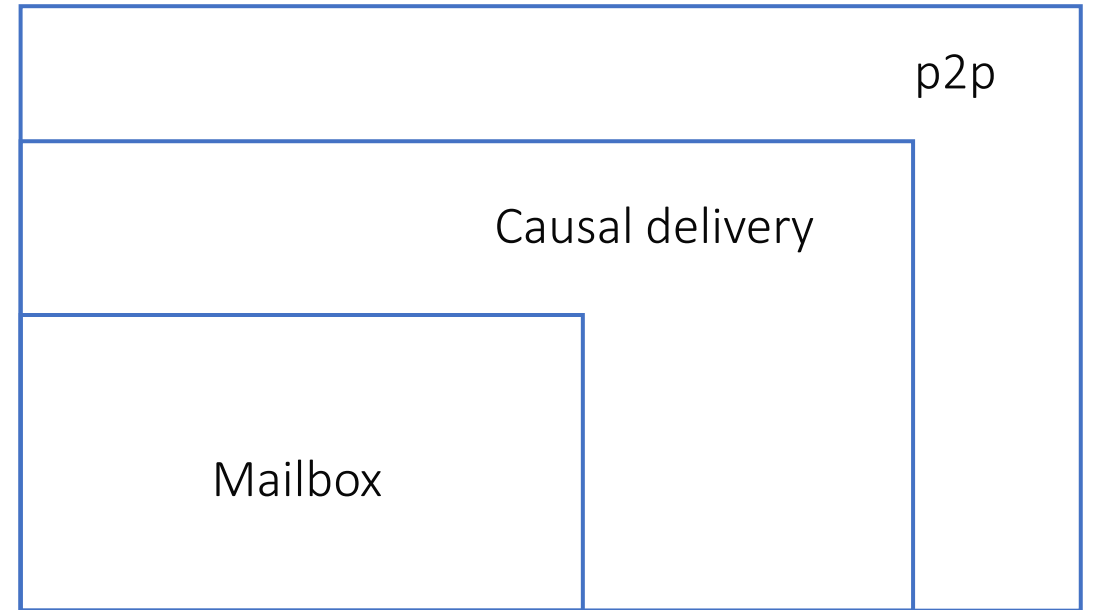
Causal delivery without taking into consideration unmatched messages

$n-1 \rightarrow \text{Causal ordering}$
 $\text{Causal ordering} \nrightarrow n-1$

MSCs hierarchy



No unmatched messages



With unmatched messages