Refinement

Mário S. Alvim
(msalvim@dcc.ufmg.br)

Quantitative Information Flow

DCC-UFMG
(2021/1)

Introduction

- Originally, the term "refinement" was used to describe a a hierarchical method of program development in which:
  1. larger, less precise descriptions of a system are "refined" into finer, more precise ones; and
  2. the processes continues until we achieve very small components that are directly implementable in some programming language or hardware.
- Nowadays, the term **refinement** is often understood in Computer Science as a relation between systems that represents preservation of properties:

  "One system S is refined by another system I, just when every property satisfied by S is satisfied by I also."

- We denote refinement by

$$S \sqsubseteq I \ ,$$

  and we can understand it informally as:

  "If a specification S is 'good enough', and an implementation I refines it, then we can be sure that I is good enough too."

Refinement:
for the customer;
for the developer

Refinement: for the *customer*; for the *developer*

- Here we'll understand refinement as a relation between the meanings of the small pieces and the larger one they replace.

  More precisely, the properties of the larger one should be preserved by the combination of small ones.

- A parameter necessary to define refinement is exactly an unambiguous description of the properties that it must preserve.

  For instance, the property to be preserved could be:

  (a) **Functionality**, so if $S \sqsubseteq I$ then any answer that I could give is an answer that S could also have given (from the same input).

  (b) **Time performance**, so execution takes no more than so-many seconds on a certain hardware configuration.

  (c) **Safety**, so a given undesirable state is guaranteed not to occur during execution (e.g., the explosion of a nuclear power plant).

  (d) **Security**, so some information leakage guarantee is satisfied.

  (e) ...

Refinement: for the *customer*; for the *developer*

- Here we'll concentrate on a refinement relation defined:
  1. wrt. "being secure" properties formalized by QIF, and
  2. where systems are represented as channels.
- Essentially, for channels S, I, we'll have

$$S \sqsubseteq I$$

  just when I is at least as secure as S.
- More precisely we'll consider the **contexts** in which security might be an issue, defined by, e.g., by a prior and gain function.

  We'll see two important ways of formulating refinement:
  - "structural refinement", which is how it's done by the developer, and
  - "testing refinement", which is how it's experienced by the customer.
- We'll also show that, suitably formulated, those two forms of refinement are exactly the same!

Structural refinement: The developer's point of view

Structural refinement: Deterministic channels (concrete)

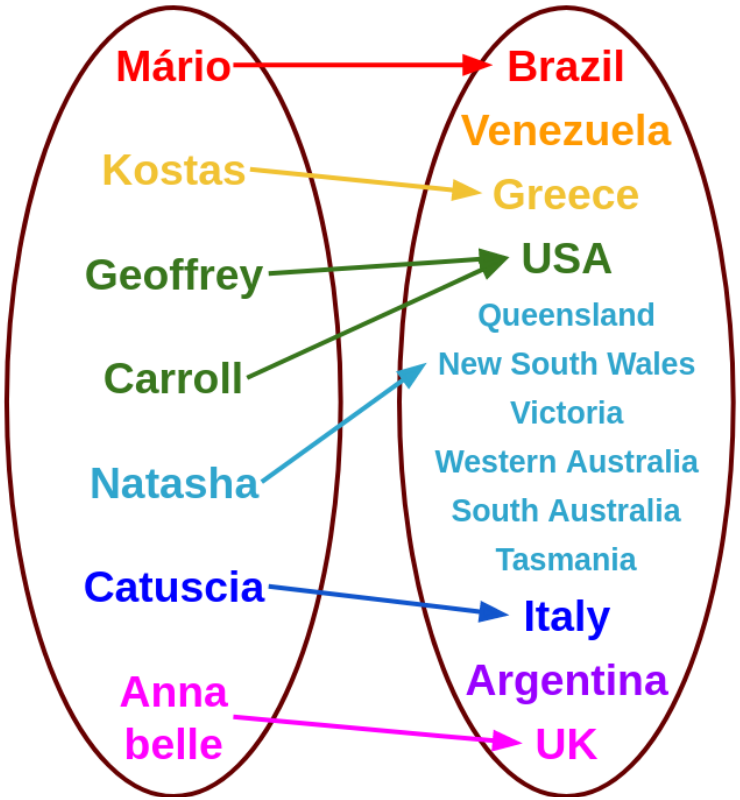
- We start our study of refinement by structural refinement, which is how it's done by the developer.
- We begin by defining structural refinement for deterministic channels.
- Recall that a channel matrix from $\mathcal{X}$ to $\mathcal{Y}$ has type $\mathcal{X} \mathrel{-\!\!\triangleright} \mathcal{Y}$.

  But, if the channel is deterministic, its type is equivalent to that of a deterministic function, i.e., $\mathcal{X} \to \mathcal{Y}$.
- A deterministic channel induces a **partition** on $\mathcal{X}$, which is a set of mutually disjoint subsets of $\mathcal{X}$ that we call **cells** (or **blocks**).
  - Inputs x_1 and x_2 in $\mathcal{X}$ belong to the same cell just when the channel maps them to the same output observation in $\mathcal{Y}$.
  - Equivalently, each cell of the partition is the pre-image of some output y.

Structural refinement: Deterministic channels (concrete)

- Example 1 Consider a deterministic channel A taking a secret person X to their location of birth, giving just the country of birth normally, but giving the state in the case of Australians.

  This deterministic function induces a partition on its input containing cells:



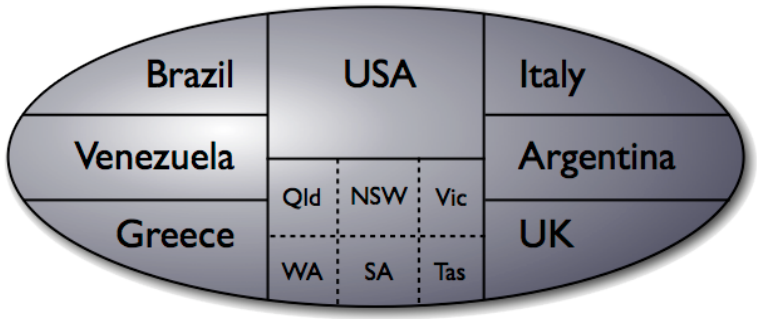
- {Mário} with all Brazilians,
- {Kostas} with all Greeks,
- {Geoffrey, Carroll} with all Americans,
- {Natasha} with all Australians from NSW,
- {Catuscia} with all Italians,
- {Annabelle} with all British,
- 3 empty cells, each with all Venezuelans, Argentinians, and Australians not born in NSW.

◁

Structural refinement: Deterministic channels (concrete)

- Example 1 (Continued)

  We can represent the partition induced by channel A as

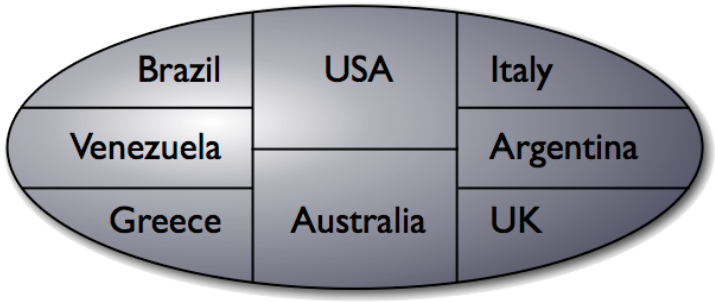


- Since we intend "is a refinement of" to mean "is at least as secure as", how can we refine deterministic channel A to another deterministic channel B?
- One condition that is intuitively sufficient is that B's partition should be "coarser" than A's partition.
  - That is, each cell of B's partition is formed by merging one or more cells of A's partition.

Structural refinement: Deterministic channels (concrete)

- Example 1 (Continued)

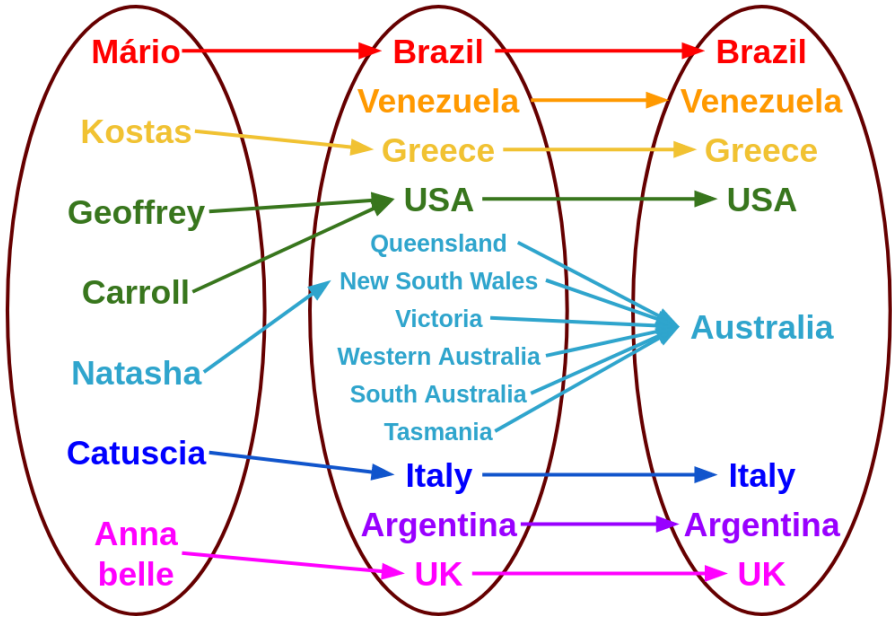
For example, a channel B that reveals only the country of birth in all cases would induce partition



.

B's partition is formed by:

- merging the cells corresponding to the six Australian states into a single cell (Australia),
- while leaving all other countries untouched.



◁

Structural refinement: Deterministic channels (concrete)

- That discussion leads to the following definition of structural refinement ($\sqsubseteq_\circ$) for deterministic channels.

Definition (9.1 - Structural refinement ($\sqsubseteq_\circ$), deterministic case)

Two deterministic channels A and B on input $\mathcal{X}$ are said to be in the **structural-refinement** relation, written A $\sqsubseteq_\circ$ B, just when the partition induced by B is coarser than that induced by A, in that each of B's cells is formed by *merging* one or more of A's cells.

- It's intuitively clear that a security-conscious "customer" will always prefer B to A, whatever:
  - the input prior π, or
  - the leakage measure might be.

  (And an adversary's preference will be exactly the opposite.)

  That's because B never gives more information about X than A would have.

Structural refinement: Deterministic channels (concrete)

- That intuition is supported by the following theorem, which says that partition coarsening is **sound** with respect to Bayes leakage at least.

Theorem (9.2)

If A *and* B *are deterministic channels and* $\mathsf{A} \sqsubseteq_{\circ} \mathsf{B}$*, then the Bayes leakage of* B *is never greater than that of* A*, regardless of the prior* π*.*

Proof. Recall from Thm. 5.15 that posterior Bayes vulnerability is simply the sum of the column maximums of the joint matrix.

In case $\mathsf{A} \sqsubseteq_{\circ} \mathsf{B}$, note that the joint matrix for B (under any prior π) is formed by merging some of the columns of the joint matrix for A, which can only decrease that sum (by converting a sum of maximums into a maximum of sums). □

Structural refinement: Deterministic channels (concrete)

- More interestingly, the converse implication holds as well. This means that partition coarsening is **complete** for Bayes leakage.

Theorem (9.3 - Deterministic coriaceous)

If A *and* B *are deterministic channels and the Bayes leakage of* B *is never greater than that of* A*, regardless of the prior* π*, then* $\mathsf{A} \sqsubseteq_{\circ} \mathsf{B}$*.*

Proof. We argue the contrapositive.

If $\mathsf{A} \not\sqsubseteq_{\circ} \mathsf{B}$, then there must exist x_1 and x_2 that belong to the same cell of A, but to different cells of B.

On a prior that gives nonzero probability only to x_1 and x_2, that is whose support is just $\{x_1, x_2\}$, channel A leaks nothing about X, while B leaks everything. □

Structural refinement: Deterministic channels (concrete)

- The Bayes leakage featuring in the two theorems above is of course a testing- , that is a customer-oriented (rather than a structural) artifact.

  The customer "experiences" with what probability his secrets can be guessed.

- We have discussed it here only to bolster the intuition.

  Soon we'll return to the two theorems above when we discuss the equivalence of structural and testing refinement in general.

Structural refinement: Probabilistic channels (concrete)

- We now consider the question of how we can generalize structural refinement

$$A \sqsubseteq_{\circ} B \ ,$$

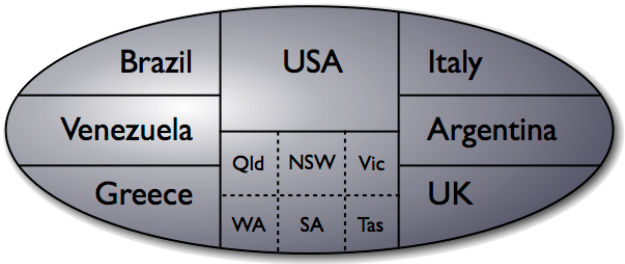
  from deterministic to probabilistic channels.

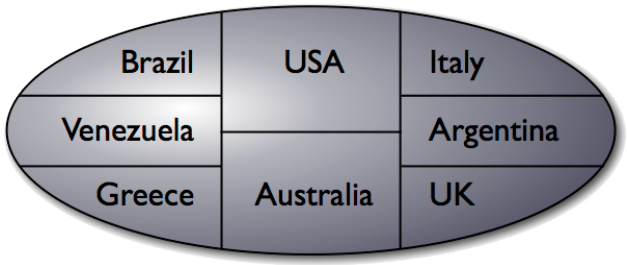
- In the deterministic case, we saw that A's partition is refined by B's just when we can convert A into B by doing a "post-processing" step in which certain of A's outputs are merged.

- The next example will revisit this definition.

Structural refinement: Probabilistic channels (concrete)

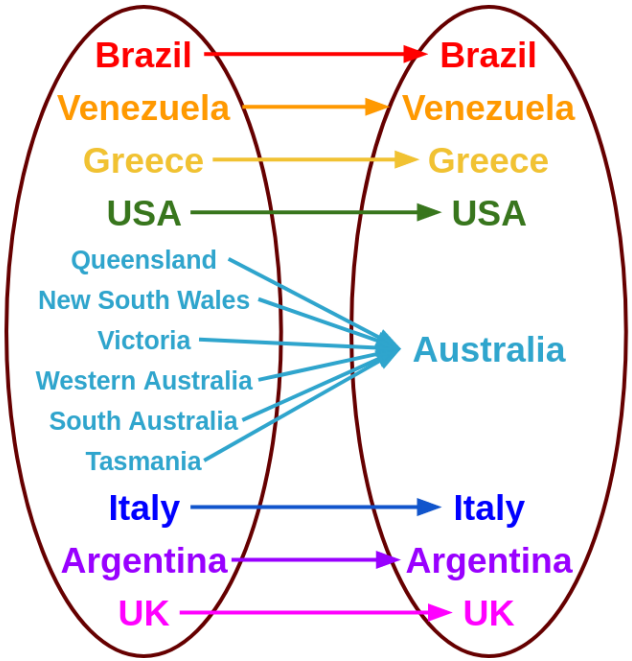
- Example 2 Let's go back to our country example, where we federated (i.e., merged) the states of Australia to

get from 

to 

.

We did that by remapping each element of the first set to an element of the second set.



Structural refinement: Probabilistic channels (concrete)

- Example 2 (Continued)

Using our matrix representation for channels, we can express merging as a cascade, so B is the matrix product of A and a channel matrix R.

$$B \quad = \quad A \cdot R \; .$$

That is, one can imagine the observables of A "pouring in" to the input of R.

In our example, the original channel A representing the mapping from people to country of birth –or state, for Austrlians– can be represented as the matrix

A	BR	VZ	GR	USA	Qld	NSW	Vic	WA	SA	Tas	IT	AR	UK
Mário	**1**	0	0	0	0	0	0	0	0	0	0	0	0
Kostas	0	0	**1**	0	0	0	0	0	0	0	0	0	0
Geoffrey	0	0	0	**1**	0	0	0	0	0	0	0	0	0
Carroll	0	0	0	**1**	0	0	0	0	0	0	0	0	0
Natasha	0	0	0	0	0	**1**	0	0	0	0	0	0	0
Catuscia	0	0	0	0	0	0	0	0	0	0	**1**	0	0
Annabelle	0	0	0	0	0	0	0	0	0	0	0	0	**1**

.

Structural refinement: Probabilistic channels (concrete)

- Example 2 (Continued)

The refinement channel R remapping country/state of birth to just country can be represented by the matrix

R	BR	VZ	GR	USA	AUS	IT	AR	UK
BR	**1**	0	0	0	0	0	0	0
VZ	0	**1**	0	0	0	0	0	0
GR	0	0	**1**	0	0	0	0	0
USA	0	0	0	**1**	0	0	0	0
Qld	0	0	0	0	**1**	0	0	0
NSW	0	0	0	0	**1**	0	0	0
Vic	0	0	0	0	**1**	0	0	0
WA	0	0	0	0	**1**	0	0	0
SA	0	0	0	0	**1**	0	0	0
Tas	0	0	0	0	**1**	0	0	0
IT	0	0	0	0	0	**1**	0	0
AR	0	0	0	0	0	0	**1**	0
UK	0	0	0	0	0	0	0	**1**

.

Structural refinement: Probabilistic channels (concrete)

- Example 2 (Continued)

And the refined channel B mapping people to just their country of birth can be represented by the matrix

B	BR	VZ	GR	USA	AUS	IT	AR	UK
Mário	**1**	0	0	0	0	0	0	0
Kostas	0	0	**1**	0	0	0	0	0
Geoffrey	0	0	0	**1**	0	0	0	0
Carroll	0	0	0	**1**	0	0	0	0
Natasha	0	0	0	0	**1**	0	0	0
Catuscia	0	0	0	0	0	**1**	0	0
Annabelle	0	0	0	0	0	0	0	**1**

.

It's easy to verify that $B = A \cdot R$.

Structural refinement: Probabilistic channels (concrete)

- Example 2 (Continued)

Indeed, we can verify that if:

- channel A represents a function $f : \mathcal{X} \to \mathcal{Y}$, and
- channel R represents a function $g : \mathcal{Y} \to \mathcal{Z}$,

then channel $\mathsf{B} = \mathsf{AR}$ represents the composite function $g \circ f : \mathcal{X} \to \mathcal{Z}$.

To see why, consider that $\mathsf{B}_{x,z}$ is given by

$$
\begin{aligned}
(\mathsf{AR})_{x,z} &= \sum_y \mathsf{A}_{x,y}\mathsf{R}_{y,z} && \text{(def. of matrix mult.)} \\
&= \mathsf{A}_{x,f(x)}\mathsf{R}_{f(x),z} && (\mathsf{A}_{x,y} \neq 0 \text{ iff } y = f(x)) \\
&= \mathsf{R}_{f(x),z} && (\mathsf{A}_{x,f(x)} = 1) \\
&= \begin{cases} 1, & \text{if } \mathsf{z} = \mathsf{g}(\mathsf{f}(\mathsf{x})), \\ 0, & \text{otherwise.} \end{cases} && \text{(R implements function } g\text{)}
\end{aligned}
$$

So we conclude that $(\mathsf{AR})_{x,z}$ represents $g \circ f$. ◁

Structural refinement: Probabilistic channels (concrete)

- The observation of the previous example can be complemented as follows.

Theorem (9.4)

Let $\mathsf{A}\colon \mathcal{X} \to\!\!\triangleright \mathcal{Y}$ *and* $\mathsf{B}\colon \mathcal{X} \to\!\!\triangleright \mathcal{Z}$ *be deterministic channel matrices.*

Then $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$ *just when there is a deterministic matrix* $\mathsf{R}\colon \mathcal{Y} \to\!\!\triangleright \mathcal{Z}$ *such that* $\mathsf{B} = \mathsf{AR}$.

Proof. If $\mathsf{B} = \mathsf{AR}$ for some deterministic R, then $\mathsf{A}(x_1) = \mathsf{A}(x_2)$ implies that $\mathsf{B}(x_1) = \mathsf{R}(\mathsf{A}(x_1)) = \mathsf{R}(\mathsf{A}(x_2)) = \mathsf{B}(x_2)$, where because the matrices are deterministic we can consider them as functions — effectively we read $\mathsf{A}\colon \mathcal{X} \to\!\!\triangleright \mathcal{Y}$ as $\mathsf{A}\colon \mathcal{X} \to \mathcal{Y}$. Hence every cell of A's partition is entirely contained in a cell of B's partition, which implies that $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$. Note that each column of R determines a cell of B, where the 1's in that column indicate which of A's cells were merged to make it.

Conversely, if $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$, then for every y we know that B maps all x's in $\mathsf{A}^{-1}(y)$ to the same value, say y'_x. If we define (deterministic) R to map each y to y'_x, then indeed $\mathsf{B} = \mathsf{AR}$. □

Structural refinement: Probabilistic channels (concrete)

- Thm. 9.4 above suggests a generalization of structural refinement to probabilistic channels: just allow A, B and R to be all probabilistic!

Definition (9.5 - Structural refinement ($\sqsubseteq_\circ$), probabilistic case)

For (probabilistic) channel matrices A and B, we say that A is structurally refined by B, again written $A \sqsubseteq_\circ B$, just when there exists a –possibly probabilistic– matrix R such that $AR = B$.

- The refinement matrix R can be considered to be a **witness** for (structural) refinement, that is a single artifact that shows the refinement to hold.

Structural refinement: Probabilistic channels (concrete)

- Note that $(\sqsubseteq_\circ)$ is:
  - **Reflexive**, meaning that every channel is refined by itself.

    Formally, every channel A satisfies

    $$A \sqsubseteq_\circ A \ .$$

    To see why, note that $A = A\,I$, where I is the identity matrix.
  - **Transitive**, meaning that if a channel A is refined by B, and B is refined by C, then A is refined by C.

    Formally, for all channels A, B and C, we have

    $$A \sqsubseteq_\circ B \quad \text{and} \quad B \sqsubseteq_\circ C \quad \Longrightarrow \quad A \sqsubseteq_\circ C \ .$$

    To see why, notice that if $B = AR$ and $C = BR'$, then we can write

    $$C \ = \ BR' \ = \ (AR)R' \ = \ A(RR') \ .$$

    (Recall that the product of two stochastic matrices is again stochastic, so RR' is itself a valid post-processing channel matrix.)

Structural refinement: Probabilistic channels (abstract)

- We note however that Def. 9.5 is in terms of concrete channels, i.e. channel matrices, whereas our main object of study is actually abstract channels.
- It's possible to give an equivalent definition in terms of abstract channels directly.

  But here we'll use a simpler "hybrid" definition that is sufficient (and equivalent).

Definition (9.6 - abstract refinement)

Let A and B be abstract channels over the same input $\mathcal{X}$.

Then A is said to be structurally refined by B, again written $A \sqsubseteq_\circ B$, just when for *all* channel matrices A and B realizing A and B (i.e. for any A, B with $[\![\mathsf{A}]\!] = A$ and $[\![\mathsf{B}]\!] = B$), we have $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$.

Structural refinement: Probabilistic channels (abstract)

- To show that the quantification over all concrete matrices makes sense in the above definition, we prove the following theorem.

Theorem (9.7)

For any channel matrix C*, we have both* $C \sqsubseteq_\circ C^r$ *and* $C^r \sqsubseteq_\circ C$ *.*

Proof. The reduced matrix C^r of channel matrix C is defined in Def. 4.9 via a series of operations: deleting all-zero columns, summing similar columns together, and ordering the resulting columns lexicographically. Each of those can be effected via post-multiplication by a simple channel matrix; and so their overall effect is achieved via multiplication by the matrix product of all those simple matrices taken together, again a channel matrix. Hence $C \sqsubseteq_\circ C^r$.

For the reverse direction the operations are adding an all-zero column, splitting a column into several similar columns, and reordering columns. Again all of those can be achieved by post-multiplication. Hence $C^r \sqsubseteq_\circ C$. □

Structural refinement: Probabilistic channels (abstract)

- **Example 3** As an illustration of the previous theorem, let channel matrix C be

C	y_1	y_2	y_3	y_4
x_1	1	0	0	0
x_2	$1/4$	$1/2$	0	$1/4$
x_3	$1/2$	$1/3$	0	$1/6$

.

The reduced matrix C^r is then

C^r		
x_1	1	0
x_2	$1/4$	$3/4$
x_3	$1/2$	$1/2$

.

Structural refinement: Probabilistic channels (abstract)

- **Example 3** (Continued)

Now notice that we have both

C^r	
x_1	1 0
x_2	$1/4$ $3/4$
x_3	$1/2$ $1/2$

$=$

C	y_1	y_2	y_3	y_4
x_1	1	0	0	0
x_2	$1/4$	$1/2$	0	$1/4$
x_3	$1/2$	$1/3$	0	$1/6$

R	
y_1	1 0
y_2	0 1
y_3	0 0
y_4	0 1

and

C	y_1	y_2	y_3	y_4
x_1	1	0	0	0
x_2	$1/4$	$1/2$	0	$1/4$
x_3	$1/2$	$1/3$	0	$1/6$

$=$

C^r	
x_1	1 0
x_2	$1/4$ $3/4$
x_3	$1/2$ $1/2$

R'	y_1	y_2	y_3	y_4
	1	0	0	0
	0	$2/3$	0	$1/3$

,

so that indeed $C \sqsubseteq_\circ C^r \sqsubseteq_\circ C$.

◁

Structural refinement: Probabilistic channels (abstract)

- Then we have the following useful result.

Corollary (9.9)

Let A and B be abstract channels over the same input $\mathcal{X}$, and let A and B be arbitrary realizations of A and B as channel matrices. Then $A \sqsubseteq_\circ B$ iff $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$.

Proof. The forward implication is immediate from Def. 9.6

For the backward implication, let A' and B' be any (other) realizations of A and B, so that by Cor. 4.11 we have $(\mathsf{A}')^r = \mathsf{A}^r$ and $(\mathsf{B}')^r = \mathsf{B}^r$. Hence, using Thm. 9.7 we have

$$\mathsf{A}' \sqsubseteq_\circ (\mathsf{A}')^r = \mathsf{A}^r \sqsubseteq_\circ \mathsf{A} \sqsubseteq_\circ \mathsf{B} \sqsubseteq_\circ \mathsf{B}^r = (\mathsf{B}')^r \sqsubseteq_\circ \mathsf{B}' \ .$$

Hence, since $(\sqsubseteq_\circ)$ is reflexive and transitive, we have $\mathsf{A}' \sqsubseteq_\circ \mathsf{B}'$. □

- Finally, we remark that we can test efficiently whether $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$, by formulating it as a linear-programming problem.

Testing refinement: The customer's point of view

Testing refinement: The customer's point of view

- We'll base our tests for refinement on the g-vulnerabilities.
- As have seen, a gain function g is intended to:
  1. capture and quantify the value $g(w, x)$ to the adversary of secrets x we are trying to protect from her, and
  2. to set out the actions w she has for trying to discover and exploit them.
- Thus we base testing refinement on what we call the **"strong g-vulnerability order"**.

Definition (9.10 - Testing refinement, probabilistic case)

Given abstract channels A and B, over the same input $\mathcal{X}$, we say that A is testing-refined by B, written $A \sqsubseteq_{\mathbb{G}} B$, if for any prior π and gain function $g\colon \mathbb{G}\mathcal{X}$ we have $V_g[\pi \triangleright A] \geq V_g[\pi \triangleright B]$.

Equivalently, we could use loss functions Def. 3.4 for the same definition, i.e. that for any loss function ℓ we have $U_\ell[\pi \triangleright A] \leq U_\ell[\pi \triangleright B]$.

Testing refinement: The customer's point of view

- Note that in Def. 9.10 we could have compared leakages rather than vulnerabilities, suggesting the name "**strong g-leakage order**".

  But the result would have been the same (regardless of whether we measure leakage multiplicatively or additively) because we have

$$\begin{array}{lrcl} & \dfrac{V_g[\pi \triangleright A]}{V_g(\pi)} & \geq & \dfrac{V_g[\pi \triangleright B]}{V_g(\pi)} \\ \text{iff} & V_g[\pi \triangleright A] & \geq & V_g[\pi \triangleright B] \\ \text{iff} & V_g[\pi \triangleright A] - V_g(\pi) & \geq & V_g[\pi \triangleright B] - V_g(\pi) \quad , \end{array}$$

  using (in the first step) the fact that g's being in $\mathbb{G}\mathcal{X}$ ensures $V_g(\pi) \geq 0$. (In the multiplicative case we assume that $V_g(\pi) \neq 0$.)

- Now that we have defined both structural and testing refinement, we'll study how they relate to each other.

Soundness of structural refinement

Soundness of structural refinement

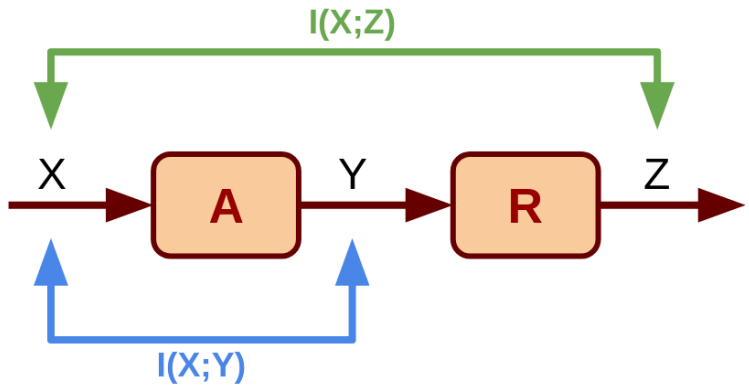
- By **"soundness"** of structural refinement we mean that
  - if structural refinement is used by a developer,
  - then he will achieve testing refinement for the customer.

  Put less formally: if he follows sound engineering practices, then he won't get sued by a litigious customer.

- The soundness of $(\sqsubseteq_\circ)$ can be understood as a generalized **data-processing inequality (DPI)** from information theory.

  The original DPI says that if $X \to Y \to Z$ is a Markov Chain, then

  $$I(X;Z) \leq I(X;Y) \ .$$



- DPI is often described with the slogan:

  "Post-processing can only destroy information."

Soundness of structural refinement

- The soundness of $(\sqsubseteq_\circ)$ is established by the following theorem, which can be understood as a generalized data-processing inequality.

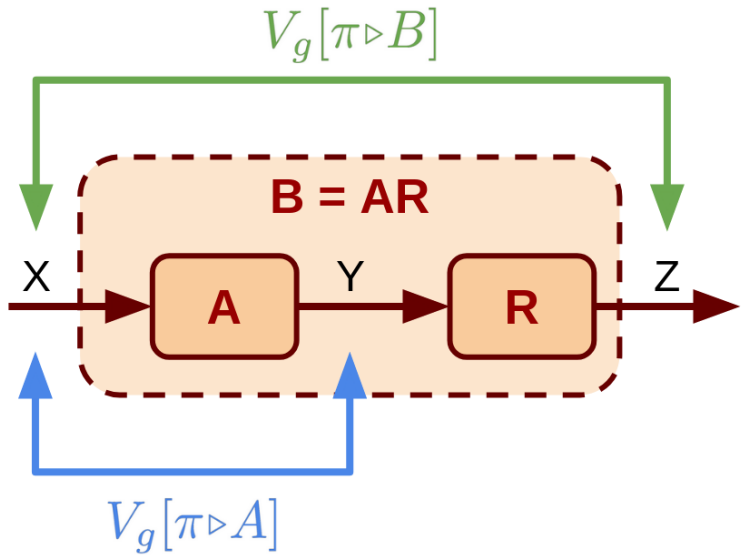
Theorem (9.11 - Soundness of structural refinement)

For abstract channels A, B over the same input space $\mathcal{X}$, we have that $A \sqsubseteq_\circ B$ implies $A \sqsubseteq_{\mathbb{G}} B$.

That is, if channel B is obtained by post-processing the output of channel A, then B can never leak more information about A's input than A itself.

For any prior π and gain function $g\colon \mathbb{G}\mathcal{X}$,

$$V_g[\pi \triangleright B] \leq V_g[\pi \triangleright A] \ .$$



Soundness of structural refinement

- **Proof.** Let abstract channels A, B (both of type $\mathbb{D}\mathcal{X}\to\mathbb{D}^2\mathcal{X}$) be realized by matrices $\mathsf{A}\colon \mathcal{X}\to\triangleright\mathcal{Y}$ and $\mathsf{B}\colon \mathcal{X}\to\triangleright\mathcal{Z}$. Recall from Cor. 9.7 that it does not matter which realizations we choose.

  Suppose therefore that $\mathsf{A} \sqsubseteq_\circ \mathsf{B}$, which means that there exists a stochastic matrix $\mathsf{R}\colon \mathcal{Y}\to\triangleright\mathcal{Z}$ such that $\mathsf{B} = \mathsf{AR}$. To show $\mathsf{A} \sqsubseteq_{\mathbb{G}} \mathsf{B}$ we must show that for any prior π and gain function g we have $V_g[\pi\triangleright\mathsf{A}] \geq V_g[\pi\triangleright\mathsf{B}]$.

  Now recall the *trace-based* formulation of posterior g-vulnerability that was established in Thm. 5.24: it is

$$\begin{aligned} V_g[\pi\triangleright\mathsf{A}] &= \max_{\mathsf{S}_A} \mathbf{tr}(\mathsf{G}^{\lceil\pi\rfloor}\mathsf{AS}_A) \quad \text{and} \\ V_g[\pi\triangleright\mathsf{B}] &= \max_{\mathsf{S}_B} \mathbf{tr}(\mathsf{G}^{\lceil\pi\rfloor}\mathsf{BS}_B) \quad , \end{aligned}$$

  where S_A and S_B are *strategies*, meaning stochastic matrices that select (possibly probabilistically) the action w an (optimal) adversary could take on each channel output. (Hence S_A is of type $\mathcal{Y}\to\triangleright\mathcal{W}$ and S_B of type $\mathcal{Z}\to\triangleright\mathcal{W}$.)

Soundness of structural refinement

- **Proof. (Continued)**

  Now observe that if S_B is a strategy for B and R is a stochastic matrix from $\mathcal{Y}$ to $\mathcal{Z}$, then RS_B is a strategy for A. Hence we can reason as follows:

$$
\begin{array}{lll}
 & V_g[\pi \triangleright \mathsf{B}] & \\
= & \max_{\mathsf{S}_B} \mathbf{tr}(\mathsf{G} \lceil \pi \rfloor \mathsf{BS}_B) & \text{“Thm. 5.24”} \\
= & \max_{\mathsf{S}_B} \mathbf{tr}(\mathsf{G} \lceil \pi \rfloor (\mathsf{AR})\mathsf{S}_B) & \text{“B} = \mathsf{AR}\text{”} \\
= & \max_{\mathsf{S}_B} \mathbf{tr}(\mathsf{G} \lceil \pi \rfloor \mathsf{A}(\mathsf{RS}_B)) & \text{“matrix multiplication is associative”} \\
\leq & \max_{\mathsf{S}_A} \mathbf{tr}(\mathsf{G} \lceil \pi \rfloor \mathsf{AS}_A) & \text{“S}_A \text{ can be RS}_B\text{”} \\
= & V_g[\pi \triangleright \mathsf{A}] \quad , & \text{“Thm. 5.24”}
\end{array}
$$

  which gives the inequality $V_g[\pi \triangleright \mathsf{A}] \geq V_g[\pi \triangleright \mathsf{B}]$, hence also the $V_g[\pi \triangleright A] \geq V_g[\pi \triangleright B]$ that we seek. □

Completeness of structural refinement: The Coriaceous theorem

Completeness of structural refinement: The Coriaceous theorem

- By **"completeness"** of structural refinement ($\sqsubseteq_\circ$) we mean that
  - if $A \not\sqsubseteq_\circ B$,
  - then there is a prior π and a gain function g such that $V_g[\pi \triangleright A] \not\geq V_g[\pi \triangleright B]$.
- The idea is that structural refinement is not more stringent than necessary:

  A failure of structural refinement implies a failure of testing refinement as well.

  Hence, structure refinement is "justified" as a development method, however expensive.
- There are two reasons why that is important.
  1. The first reason is that it gives the customer redress against shoddy or unscrupulous "developers".
  2. It gives the developer confidence that using structural refinement is really worth the effort (and the expense).

Completeness of structural refinement: The Coriaceous theorem

- We can now present completeness, the converse to Thm. 9.12, which established the data-processing inequality for g-vulnerabilities.

Theorem (9.12 - Completeness of structural refinement: Coriaceous)

For abstract channels A, B over the same input space $\mathcal{X}$, we have that $A \sqsubseteq_{\mathbb{G}} B$ implies $A \sqsubseteq_{\circ} B$.

- The theorem says (by its counter-positive) that
  - if B does not refine A,
  - then there exist a prior π and gain function $g\colon \mathbb{G}\mathcal{X}$ st.

$$V_g[\pi \triangleright B] > V_g[\pi \triangleright A] \ .$$

As a consequence, if B is not a refinement of A, there is for sure a context in which replacing A with B increases leakage.

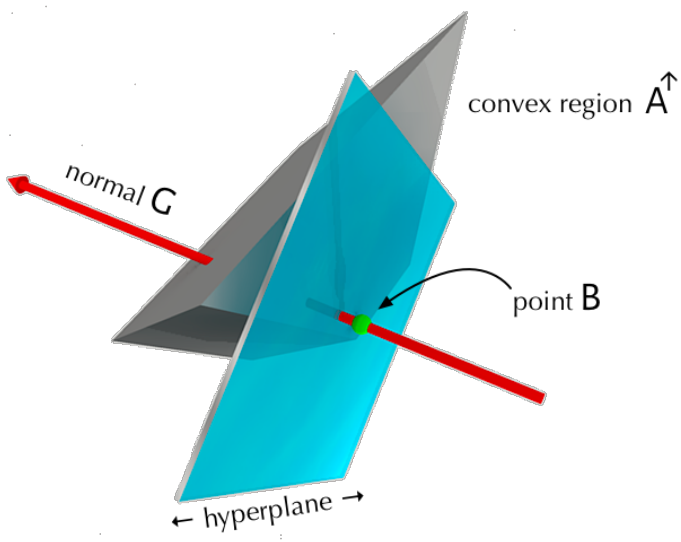
Completeness of structural refinement: The Coriaceous theorem

- **Proof.** The complete proof is given in the textbook; here's its intuition.

1. Let channel matrices A, B be realizations of channels A, B, respectively.
2. The set $\mathsf{A}^{\uparrow} = \{\mathsf{AR} \mid \mathsf{R} \text{ is a stochastic matrix from } \mathcal{Y} \text{ to } \mathcal{Z}\}$ can be represented as a convex and closed polytope in a suitable space.
3. By counter-positive, assume $A \not\sqsubseteq_{\circ} B$, so $\mathsf{B} \notin \mathsf{A}^{\uparrow}$.
4. By the Separation Hyperplane Lemma, there's a hyperplane separating A from B.
5. We can then use the normal of this hyperplane to build a gain function g that makes

$$V_g[\vartheta \triangleright \mathsf{A}] \quad < \quad V_g[\vartheta \triangleright \mathsf{B}] \ ,$$

showing that B leaks more than A under the uniform prior ϑ.



6. From that we conclude that $A \not\sqsubseteq_{\mathbb{G}} B$. □

Completeness of structural refinement: The Coriaceous theorem

- From Thm. 9.11 (soundness of $\sqsubseteq_\circ$ wrt. $\sqsubseteq_{\mathbb{G}}$) and Thm. 9.12 (completeness of $\sqsubseteq_\circ$ wrt. $\sqsubseteq_{\mathbb{G}}$) together, we have the equivalence of our two refinement orders.

Theorem (9.13 - Equivalence of structural and testing refinement orders)

The structural and the testing refinement orders on channels are the same: that is we have $(\sqsubseteq_\circ) = (\sqsubseteq_{\mathbb{G}})$.

- Hence, we are justified in writing just "$\sqsubseteq$" from now on for the refinement order between channels, whether abstract or concrete, and whether structure or tests are used.
- Note that we can efficiently test whether $A \sqsubseteq B$ by testing whether $A \sqsubseteq_\circ B$.

  In contrast, testing $A \sqsubseteq_{\mathbb{G}} B$ naïvely would require considering all priors and all gain functions!

The structure of abstract channels under refinement

The structure of abstract channels under refinement

- Recall that refinement ($\sqsubseteq$) is reflexive and transitive on channel matrices.
- Now let's show that refinement is not **antisymmetric**, so

$$A \sqsubseteq B \qquad \text{and} \qquad B \sqsubseteq A \qquad \text{does not imply} \qquad A = B \ .$$

- To see why, recall that it's possible for two concrete channels $C \neq D$ to be different, but have the same reduced channel $C^r = D^r$.

  In this case, we have by Thm. 9.7 that

$$C \sqsubseteq C^r \sqsubseteq C \qquad \text{and} \qquad D \sqsubseteq D^r \sqsubseteq D \ ,$$

  from which we conclude (using that $C^r = D^r$) that

$$C \sqsubseteq C^r = D^r \sqsubseteq D \qquad \text{and} \qquad D \sqsubseteq D^r = C^r \sqsubseteq C \ ,$$

  even if the (concrete) channel matrices C and D are different.

The structure of abstract channels under refinement

- On abstract channels, however, refinement is indeed antisymmetric, making it a **partial order**.

Theorem (9.14)

The refinement relation ($\sqsubseteq$) is antisymmetric on abstract channels.

Proof. Given in the textbook. □

Refinement and monotonicity

Refinement and monotonicity: Compositionality for contexts

- As have discussed **compositionality** is the principle that

  "The meaning of a compound is determined by the meanings of its components."

- Recall that we take the meaning of a channel matrix C to be the abstract channel $[\![\mathsf{C}]\!]$ that it denotes.

- Then we recall that compositionality of our abstraction $[\![-]\!]$ wrt. the operation of parallel composition requires that

$$[\![\mathsf{C} \,|||\, \mathsf{D}]\!] \quad = \quad [\![\mathsf{C}]\!] \,\|\, [\![\mathsf{D}]\!] \;,$$

  where $(|||)/(\|)$ are the concrete/abstract versions of parallel composition.

- Generalizing the above slightly, we could regard

  "run in parallel with D *or D"*

  as a **context** into which some C or C is deployed.

Refinement and monotonicity: Compositionality for contexts

- Given the context "run in parallel with D/D", we could then
  - write the corresponding syntactic context as $(-|||\mathrm{D})$, and call it $\boldsymbol{\mathcal{P}}_{\mathrm{D}}(-)$; and
  - write the corresponding semantic context as $(-\|D)$, and call it $\mathcal{P}_D(-)$.
- Then compositionality wrt. contexts would be in this case (dropping subscripts and parentheses) that

$$[\![\boldsymbol{\mathcal{P}}\mathrm{C}]\!] \quad = \quad \mathcal{P}[\![\mathrm{C}]\!] \,,$$

from which we can see that

$$[\![\mathrm{C}]\!] = [\![\mathrm{C}']\!] \qquad \Longrightarrow \qquad [\![\boldsymbol{\mathcal{P}}\mathrm{C}]\!] = [\![\boldsymbol{\mathcal{P}}\mathrm{C}']\!] \,.$$

That is, if you swap one component C for another C' of the same meaning (even if differently written), the meaning of the compound $\boldsymbol{\mathcal{P}}(-)$ in which the components occur is unaffected.

Refinement and monotonicity: Compositionality for contexts

- Recall that we write $C \equiv C'$ for $[\![C]\!]=[\![C']\!]$.

  Hence, the above is equivalently to

$$C \equiv C' \qquad \Longrightarrow \qquad \mathcal{P}C \equiv \mathcal{P}C' \ .$$

Refinement and monotonicity: Monotonicity with respect to refinement

- Now monotonicity is easy to define, and it applies to the concrete- and the abstract levels separately. Here we'll give it for the concrete level.

Definition (9.16 - Concrete monotonicity)

A concrete context $\mathcal{C}$ is monotonic for refinement just when for all concrete channels C, D we have

$$\mathsf{C} \sqsubseteq \mathsf{D} \quad \text{implies} \quad \mathcal{C}(\mathsf{C}) \sqsubseteq \mathcal{C}(\mathsf{D}) \quad .$$

- Without a monotonicity property at the concrete level, rigorous software engineering is very difficult — perhaps impossible!
- Compositionality matters because security protocols are not used in a vacuum — but often they are analyzed in a vacuum.

Even if we are forced to analyze in a vacuum, we'd better make sure we use an analysis method where the conclusions still hold when we deploy the mechanism back in "the atmosphere"!

Why does refinement ($\sqsubseteq$) have to be so complicated?

Why does refinement ($\sqsubseteq$) have to be so complicated?

- As a quick remark, in the book we provide several reasons to explain‘ why refinement has to be so complicated.
  - For now we'll assume that arguments based on the strong g-leakage ordering are enough compelling evidence.
  - But when we talk about Dalenius scenarios, we'll introduce a justification for refinement based on "compositional closure".

Capacity is unsuitable as a criterion for refinement

Capacity is unsuitable as a criterion for refinement

- Still in line with the question of why refinement has to be so complicated, we may ask ourselves the following.

  "Could an alternative to the definition of testing-refinement ($\sqsubseteq_{\mathbb{G}}$ of Def. 9.10) use capacities?"

- That question makes sense, since capacities can be made independent of the prior and the gain function, depending on which kind of capacity is used.

  In particular, recall from the Miracle theorem (Thm. 7.5) that multiplicative Bayes capacity is an upper bound on multiplicative g-leakage for all π and $g\colon \mathbb{G}^{+}\mathcal{X}$.

Capacity is unsuitable as a criterion for refinement

- Hence, one might try to define refinement as

$$A \sqsubseteq_{\mathbb{G}} B \qquad \stackrel{?}{\equiv} \qquad \mathcal{ML}_1^{\times}(\mathbb{D}, A) \geq \mathcal{ML}_1^{\times}(\mathbb{D}, B) \ ,$$

  under the understanding that, from the customer's point of view, B should be regarded as "more secure" than A if its worst-case multiplicative leakage is smaller.

- But an ordering on maximums does not imply an ordering on individuals!
  - There could be particular π's and g's where the leakage of that B is, in fact, greater than that of the A.
  - If such a pair characterizes the context in which our system is deployed, then we might well regret replacing A with B.

Capacity is unsuitable as a criterion for refinement

- **Example 4** Consider the two channel matrices

A	z_1	z_2	z_3
x_1	$3/4$	$1/8$	$1/8$
x_2	$1/8$	$3/4$	$1/8$
x_3	$1/8$	$1/8$	$3/4$

and

B	y_1	y_2	y_3
x_1	1	0	0
x_2	0	$1/2$	$1/2$
x_3	0	$1/2$	$1/2$

.

Suppose we compare A and B wrt. Bayes capacity.

Recall that for any fixed π, we have established that

$$\mathcal{L}_1^{\times}(\pi, \mathrm{A}) \;\geq\; \mathcal{L}_1^{\times}(\pi, \mathrm{B}) \qquad \text{iff} \qquad \mathcal{L}_1^{+}(\pi, \mathrm{A}) \;\geq\; \mathcal{L}_1^{+}(\pi, \mathrm{B}) \quad .$$

Hence we might expect that it makes no difference whether we compare multiplicative or additive Bayes capacity...

But this is not so!

Capacity is unsuitable as a criterion for refinement

- Example 4 (Continued)

Let's first consider the multiplicative Bayes capacity of channels

A	z_1	z_2	z_3
x_1	$3/4$	$1/8$	$1/8$
x_2	$1/8$	$3/4$	$1/8$
x_3	$1/8$	$1/8$	$3/4$

and

B	y_1	y_2	y_3
x_1	1	0	0
x_2	0	$1/2$	$1/2$
x_3	0	$1/2$	$1/2$

.

Using Thm. 7.2 we have

$$\mathcal{ML}_1^{\times}(\mathbb{D}, \mathsf{A}) = 3/4 + 3/4 + 3/4 = 9/4 \quad \text{and}$$

$$\mathcal{ML}_1^{\times}(\mathbb{D}, \mathsf{B}) = 1 + 1/2 + 1/2 = 2 \ ,$$

so

$$\mathcal{ML}_1^{\times}(\mathbb{D}, \mathsf{A}) > \mathcal{ML}_1^{\times}(\mathbb{D}, \mathsf{B}) \ .$$

Capacity is unsuitable as a criterion for refinement

- Example 4 (Continued)

Now let's consider additive Bayes capacity of channels

A	z_1	z_2	z_3
x_1	$3/4$	$1/8$	$1/8$
x_2	$1/8$	$3/4$	$1/8$
x_3	$1/8$	$1/8$	$3/4$

and

B	y_1	y_2	y_3
x_1	1	0	0
x_2	0	$1/2$	$1/2$
x_3	0	$1/2$	$1/2$

.

By trying all sub-uniform distributions, we find that:

- $\mathcal{ML}_1^+(\mathbb{D}, \mathsf{A})$ is realized on the uniform prior $\vartheta = (1/3, 1/3, 1/3)$ and

$$\mathcal{ML}_1^+(\mathbb{D}, \mathsf{A}) \quad = \quad V_1[\vartheta \triangleright \mathsf{A}] - V(\vartheta) \quad = \quad 3/4 - 1/3 \quad = \quad 5/12 \quad .$$

- $\mathcal{ML}_1^+(\mathbb{D}, \mathsf{B})$ is realized on a non-uniform prior $\pi = (1/2, 1/2, 0)$ and

$$\mathcal{ML}_1^+(\mathbb{D}, \mathsf{B}) \quad = \quad V_1[\pi \triangleright \mathsf{B}] - V(\pi) \quad = \quad 1 - 1/2 \quad = \quad 1/2 \quad .$$

So

$$\mathcal{ML}_1^+(\mathbb{D}, \mathsf{A}) \quad < \quad \mathcal{ML}_1^+(\mathbb{D}, \mathsf{B}) \; .$$

Capacity is unsuitable as a criterion for refinement

- Example 4 (Continued)

Hence we have

$$\begin{array}{lcll} \mathcal{ML}_1^{\times}(\mathbb{D},\mathsf{A}) & > & \mathcal{ML}_1^{\times}(\mathbb{D},\mathsf{B})\ , & \text{but} \\ \mathcal{ML}_1^{+}(\mathbb{D},\mathsf{A}) & < & \mathcal{ML}_1^{+}(\mathbb{D},\mathsf{B})\ . & \end{array}$$

These two inequalities reflect the behavior of A and B on different priors.

- The first reflects that on prior

$$\boldsymbol{\vartheta} = (1/3, 1/3, 1/3)$$

  A's multiplicative and additive Bayes leakage exceeds B's.

- The second reflects that on prior

$$\pi = (1/2, 1/2, 0)\ ,$$

  B's multiplicative and additive Bayes leakage exceeds A's.

Hence, using capacities as a notion of refinement would not necessarily give the customer the robustness she might desire. ◁