A demonic lattice of information

Carroll Morgan

University of New South Wales, and Data61; Australia.

Abstract. Landuaer and Redmond's *Lattice of Information* was an early and influential formalisation of the pure structure of security [8]: a partial order was defined for information-flow from a hidden state. In modern terms we would say that more-security *refines* less-security. For Landauer, the deterministic case [op. cit.], the refinement order is a lattice. Very recently [9, 3] a similar approach has been taken to purely probabilistic systems and there too a refinement order can be defined; but it is not a lattice [12].
In between deterministic and probabilistic is demonic, where behaviour is not deterministic but also not quantifiable. We show that our own earlier approach to this [15, 16] fits into the same pattern as deterministic and probabilistic, and illustrate that with results concerning compositionality, testing, soundness and completeness. Finally, we make some remarks about source-level reasoning.

1 A *deterministic* lattice of information — the original

1.1 Historical introduction and intuition

Landauer and Redmond proposed in 1993 *A Lattice of Information* [8] for deterministic channels that accept hidden input and produce visible output. The "information" in Landauer's title is what the channel's output tells an observer about the input that we are trying to hide from her. [1]

Definition 1. *Deterministic channel* ——— Given non-empty *input space* $\mathcal{I}$ and *output space* $\mathcal{O}$, a *deterministic channel* is a total function from $\mathcal{I}$ to $\mathcal{O}$. For channel $C{:}\,\mathcal{I}{\rightarrow}\mathcal{O}$, an input i in $\mathcal{I}$ produces an output $C(i)$ in $\mathcal{O}$. $\Box$

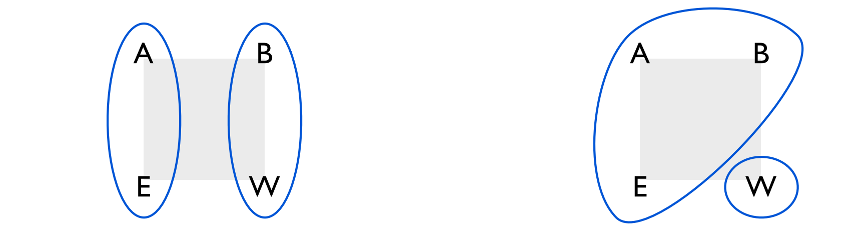
With "deterministic" we emphasise that for any input i the channel C always outputs the same output o, that is $o = C(i)$.

Take for the input space $\mathcal{I}$ the letters $\{\mathsf{A}, \mathsf{B}, \mathsf{E}, \mathsf{W}\}$, and let the output space $\mathcal{O}^1$ be $\{\textsc{vowel}, \textsc{cons}\}$ for "vowel" or "consonant"; then define channel $C^1{:}\,\mathcal{I}{\rightarrow}\mathcal{O}^1$ in the obvious way. Define another channel $C^2{:}\,\mathcal{I}{\rightarrow}\mathcal{O}^2$ where $\mathcal{O}^2$ is $\{\textsc{early}, \textsc{late}\}$ for "early" or "late" in the alphabet. These two channels $C^{1,2}$ have different output spaces $\mathcal{O}^{1,2}$ (but the same input space) because they are observing different

[1] We use the feminine she/her consistently for adversaries. Plural we/us is used for the designers or users of programs, or the readers of this article; and neuter "it" or plural "they" is used for third parties.

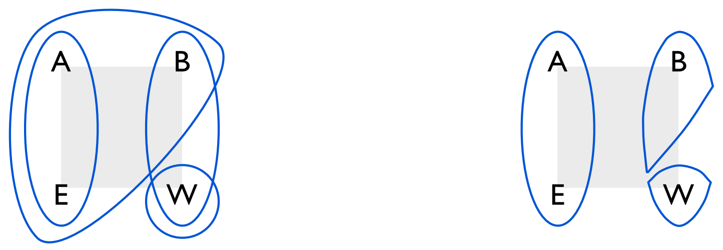
things. We compare them therefore only wrt. the information they release about their inputs: the precise values of their outputs will be seen to be irrelevant.

Each channel induces a partition on $\mathcal{I}$ via the kernels of the functions $C^{1,2}$, as shown in Fig. 1, where the partitions' cells show just which elements of $\mathcal{I}$ can be distinguished by an observer who sees the output of the channel: two input elements can be distinguished by an observer just when they are not in the same cell. Thus Fig. 1(a) shows that B, W cannot be distinguished by an observer of C^1's output, because they are both consonants; but Fig. 1(b) shows that B, W can be distinguished by C^2, because B is early but W is late.



(a) VOWEL/CONS partition for C^1 (b) EARLY/LATE partition for C^2

Fig. 1. Partitions induced on $\mathcal{I}$ by the channels C^1 and C^2



(a) Both cookie-cutters applied (b) The meet is finer than both.

Fig. 2. Induced partitions

In general we write $\mathbb{E}\mathcal{I}$ for the set of partitions of $\mathcal{I}$; clearly $\mathbb{E}\mathcal{I}$ is a subset of the powerset $\mathbb{P}\mathcal{I}$ of $\mathcal{I}$, and there is partial order that relates two partitions in $\mathbb{E}\mathcal{I}$ just when one can be formed from the other by dividing cells into smaller pieces (or, in the opposite direction, by merging cells). It is a lattice because both meet (greatest lower bound) and join (least upper bound) are defined. The meet can be visualised by thinking of partitions as "cookie cutters", with set $\mathcal{I}$ being the "dough" and *both* partitions applied, one on top of the other: the pieces formed by both cookie-cuts together determine the cells of the meet, shown for C^1 and C^2 in Fig. 2. It is the least informative channel that reveals as much about the input as each of $C^{1,2}$ does: for example the meet must distinguish A,B because C^1 does, and it must distinguish B,W because C^2 does. Complementarily, the join is the most informative channel that reveals no more than $C^{1,2}$ and is shown in Fig. 3: in this case it is the channel that reveals nothing. [2] Note that none of these constructions –neither the partial order, nor the meet/join– require details of output sets $\mathcal{O}^{1,2}$; only the partitions induced by the channels are important.

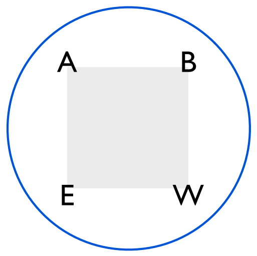


Fig. 3. The join is coarser than both $C^{1,2}$.

In software engineering the refinement order relates specifications S to implementations P just when P is as good as or better than S according to precise criteria set by the user. The available criteria are determined carefully, even "legally", beforehand and can be seen as the terms of reference available for writing contracts between user and supplier. Normally, program-refinement is written $S \sqsubseteq P$, that specification S is *refined by* implementation P. Since here our focus is on security, we consider "revealing less" to be better than "revealing more", so that we would write (exhaustively) for the examples above

$$\begin{array}{ll}
\text{Fig. 2(b)} \sqsubseteq \text{Fig. 1(a)} & \text{Fig. 1(a)} \sqsubseteq \text{Fig. 3} \\
\text{Fig. 2(b)} \sqsubseteq \text{Fig. 1(b)} & \text{Fig. 1(b)} \sqsubseteq \text{Fig. 3} \\
\text{Fig. 2(b)} = \text{Fig. 1(a)} \sqcap \text{Fig. 1(b)} & \text{Fig. 3} \quad = \text{Fig. 1(a)} \sqcup \text{Fig. 1(b)} \ .
\end{array}$$

This shows (unfortunately) that the mathematical term "partition refinement" (finer cells) and the computer-science term "program refinement" (fewer distinctions made, less information revealed, better for the user) go exactly in opposite directions. We follow the Computer-Science convention.

1.2 Definition of secure refinement for channels

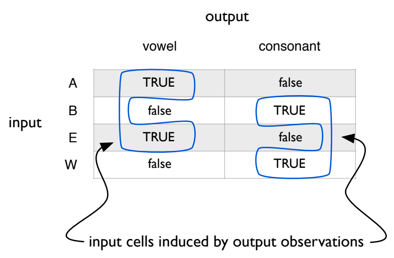
The definition of secure refinement for deterministic channels is that for $S{:}\,\mathcal{I}{\rightarrow}\mathcal{O}^S$ and $P{:}\,\mathcal{I}{\rightarrow}\mathcal{O}^P$ we have $S{\sqsubseteq}P$ just when there is a refinement-function $R{:}\,\mathcal{O}^S{\rightarrow}\mathcal{O}^P$

[2] Although both C^1 and C^2 distinguish A,W, their join cannot. Because C^2 does not distinguish A,B, the join cannot; it can't distinguish B,W because C^1 does not: by transitivity therefore the join must regard A,W as equal. The same applies to E,W.

such that $P{=}R{\circ}S$. The relation ($\sqsubseteq$) is reflexive and transitive (obviously), and it is antisymmetric. Note that channels can be in refinement even when their output spaces differ.

The intuition for its definition is that having such an R means that P's output cannot tell you anything you do not already know, at least implicitly, from the output of S: that is if you know o^S (i.e. $S(i)$), then you do not need to run P to know o^P as well (i.e. $P(i)$) — it is simply $R(o^S)$, i.e. determined by the result o^S you already have, precisely because $P{=}R{\circ}S$.

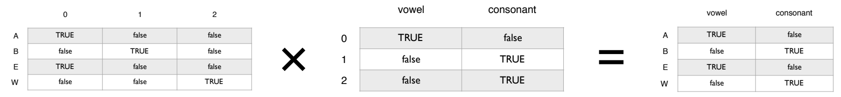
An equivalent formulation of refinement is to see S, P, R as Boolean matrices, with for example $S_{i,o} = \mathit{true}$ just when $S.i{=}o$, as in Fig. 4. [3] Because the channels S, P are deterministic (and total) the corresponding matrices have exactly one *true* in each row, and the induced cells are given by the matrix columns, with *true* entries identifying members of the cell corresponding to that column. Similarly because R represents a function, it too has exactly one *true* in each row. With matrices the formulation of refinement is that if channel S is a Boolean $\mathcal{I}{\times}\mathcal{O}^S$ matrix and channel P is a Boolean $\mathcal{I}{\times}\mathcal{O}^P$ matrix then $S{\sqsubseteq}P$ just when there is a Boolean $\mathcal{O}^S{\times}\mathcal{O}^P$ matrix R such that $P{=}SR$. [4]



	vowel	consonant
A	TRUE	false
B	false	TRUE
E	TRUE	false
W	false	TRUE

Fig. 4. Matrix representation of a deterministic channel

Fig. 5 shows how the meet of $C^{1,2}$ in Fig. 2(b) is refined in this style, i.e by a matrix R to C^1. Notice that the column-labels $0, 1, 2$ of the meet (and the rows of the refinement matrix, in the middle) have no external significance: this emphasises that it is the partition of the input cells, alone, that indicates the information flow induced by a channel. Fig. 6 uses a different matrix R' to refine the same meet to C^2 instead.

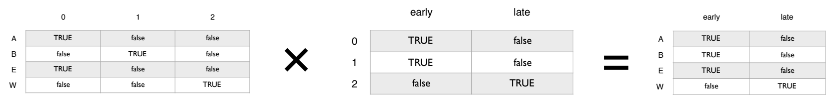


$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline A & \text{TRUE} & \text{false} & \text{false} \\ B & \text{false} & \text{TRUE} & \text{false} \\ E & \text{TRUE} & \text{false} & \text{false} \\ W & \text{false} & \text{false} & \text{TRUE} \end{array} \times \begin{array}{c|cc} & \text{vowel} & \text{consonant} \\ \hline 0 & \text{TRUE} & \text{false} \\ 1 & \text{false} & \text{TRUE} \\ 2 & \text{false} & \text{TRUE} \end{array} = \begin{array}{c|cc} & \text{vowel} & \text{consonant} \\ \hline A & \text{TRUE} & \text{false} \\ B & \text{false} & \text{TRUE} \\ E & \text{TRUE} & \text{false} \\ W & \text{false} & \text{TRUE} \end{array}$$

The column labels 0,1,2 for the matrix representing $C^1 \sqcap C^2$ are chosen arbitrarily.

Fig. 5. Refinement of $C^1 \sqcap C^2$ to C^1.

[3] For matrix M indexed by r, c we write $M_{r,c}$ for the value in row r and column c.
[4] We write SR for the matrix multiplication of S and R.



$$\begin{array}{c|ccc} & 0 & 1 & 2 \\ \hline A & \text{TRUE} & \text{false} & \text{false} \\ B & \text{false} & \text{TRUE} & \text{false} \\ E & \text{TRUE} & \text{false} & \text{false} \\ W & \text{false} & \text{false} & \text{TRUE} \end{array} \times \begin{array}{c|cc} & \text{early} & \text{late} \\ \hline 0 & \text{TRUE} & \text{false} \\ 1 & \text{TRUE} & \text{false} \\ 2 & \text{false} & \text{TRUE} \end{array} = \begin{array}{c|cc} & \text{early} & \text{late} \\ \hline A & \text{TRUE} & \text{false} \\ B & \text{TRUE} & \text{false} \\ E & \text{TRUE} & \text{false} \\ W & \text{false} & \text{TRUE} \end{array}$$

Fig. 6. Refinement of $C^1 \sqcap C^2$ to C^2.

1.3 Testing, soundness and completeness

In §1.2 the refinement function, eqv. matrix, is a witness to the refinement $S{\sqsubseteq}P$, showing not only that the partition cells induced by S can be merged to form the cells of P, but actually how to do it; the existence of such an R is in fact refinement's definition. In principle this gives a method for constructing implementations P from specifications S, a "security by design" approach where suitable matrices R guide the programmer's efforts.

The complementary problem however is how a customer should convince a court that $S{\not\sqsubseteq}K$, that when he bought S but got K he was cheated. [5] It's not practical to go through all the (infinitely many) potential R matrices and show the court, for each one, that $P{\neq}SR$. [6] Just as R provides a witness for $(\sqsubseteq)$, we need a witness for $(\not\sqsubseteq)$ too.

In this deterministic setting a witness for $(\not\sqsubseteq)$ is a subset ι of $\mathcal{I}$ such that some cell of K is a subset of ι but no cell of S is a subset of ι. Intuitively this means that there is a "Could we have let slip that i is an ι?" test that K would *fail* by revealing some cell $\kappa{\subseteq}\iota$, since K cannot release κ unless $i{\in}\kappa$. Because no cell σ of S satisfies $\sigma{\subseteq}\iota$, that slip was excluded by the specificaion.

These two witnesses, refinement matrix R for $(\sqsubseteq)$ and subset ι of $\mathcal{I}$ for $(\not\sqsubseteq)$, are related by "soundness" and "completeness". *Soundness* says that whenever $S{\sqsubseteq}X$, i.e. there exists a suitable witness R for refinement, then there cannot be any refuting witness ι that (inconsistently) would establish $S{\not\sqsubseteq}X$. In intuitive terms, it is that if a software engineer follows sound practices then he will never lose a court case. *Completeness* says that whenever $S{\not\sqsubseteq}X$, i.e. there is no refinement witness R, then there exists a refuting witness ι of that; we do not have to try (and reject) every single R.

Another way of looking at soundness and completeness in practical terms is as follows. On the one hand we want $(\sqsubseteq)$ to be weak, since the weaker it is the more implementation strategies are available to the engineer and the less he will charge for his work. But it cannot be *too* weak, e.g. the universal relation, since then he could build faulty implementations: this is soundness. On the other hand, we want $(\sqsubseteq)$ to be strong, since the stronger it is the less likely it is that implementations will disappoint their customers. But it cannot be *too* strong, e.g. the identity, since in that case the engineer will have so few design options that his products will be expensive or even impracticable. This is completeness.

[5] The mnemonics are S for S̲pecification and P for an imP̲lementation (or P̲rogram) that is supposed to refine S, and K for a "K̲ludge" that, as an example, in fact does not refine S. In uncommitted cases, neither P nor K, we will use X.

[6] In the probabilistic case §2, there would be infinitely many R's to check: it would literally take forever.

And finally, we can think of witness R as a method of construction, whereas ι is a method of testing. Follow R and our implementation P is guaranteed at least as secure as the specification S (soundness); but if we simply dream-up (i.e. cobble together) a product without such an R, and in fact there turns out not to be one, then there will be a test ι that S will pass but K will fail (completeness). And we might meet that ι in a courtroom.

2 A probabilistic partial order (not lattice) of information

The probabilistic analogue of the deterministic case §1 is communication channels with probabilistic transmission [17]. Here the input is a message to be sent and the output is the message received, chosen from a distribution determined by the input. The traditional representation of such channels is stochastic matrices where, real numbers in each row i give for each column o the probability that input i will result in output o. Deterministic channels are special cases of probabilistic channels, where *true* is probability 1 and *false* is probability 0.

As shown elsewhere [9, 10, 3, 14, 1, 11, 13, 2, 5] and mentioned above, there is a probabilistic analogue of secure refinement $S{\sqsubseteq}P$ that can be formulated as a generalisation of §1.2: the refinement matrix R is now stochastic, representing a "probabilistic merge" of S-outputs to P-outputs mediated by R such that again $P{=}SR$. This relation (between matrices) is reflexive and transitive (obviously). But it is not antisymmetric: for that we quotient to *abstract channels* where all-zero columns are removed, similar columns are merged and the order of (the remaining) columns is ignored [14]. [7] Unfortunately the resulting abstracted partial order is not a lattice [12] but, aside from that, it shares many structural properties with deterministic refinement. In particular there are probabilistic analogues of soundness and completeness, with tests based on "gain functions" over $\mathcal{I}$ which are more general than the subsets of $\mathcal{I}$ that suffice for deterministic channels [9, 3]. [8]

3 A demonic lattice of information

3.1 Basic structure

With §1 and §2 as motivating examples, we now treat our main topic: the demonic case, where observations are not necessarily wholly determined by the inputs, but we have no quantitative information about possible variations. This was earlier proposed in [15, 16], but the probabilisitic model §2 was not known (we believe) at that time.

[7] The identity matrix is stochastic, and the product of two stochastic matrices is stochastic. Matrix columns are *similar* just when each is a multiple of the other. Column order can then be ignored by representing the matrix as a set of (the remaining) columns.

[8] The completeness property was called the *Coriaceous Conjecture* in [3]. It was proved in [9, 1] and, it turns out, earlier by [4].

Definition 2. *Demonic channel: matrix formulation* —— A demonic channel from $\mathcal{I}$ to $\mathcal{O}$ is a Boolean matrix with $\mathcal{I}$-indexed rows and $\mathcal{O}$-indexed columns in which each row has at least one *true* element. □

Whereas deterministic channels induce partitions on their input-space $\mathcal{I}$, demonic channels induce more generally simply sets of subsets of $\mathcal{I}$, i.e. like partitions but allowing the cells to overlap. The overlaps occur just for those i-rows containing more than one *true*: those shared i's "belong" more to than one o-column, i.e. to more than one cell. [9] We now give a more abstract definition in those terms.

Definition 3. *Secure refinement for demonic channels: matrix formulation*
A demonic "specification" channel $S{:}\,\mathcal{I}{\twoheadrightarrow}\mathcal{O}^S$ is secure-refined by an "implementation" channel $P{:}\,\mathcal{I}{\twoheadrightarrow}\mathcal{O}^P$ just when there is a demonic matrix $R{:}\,\mathcal{O}^S{\twoheadrightarrow}\mathcal{O}^P$ such that $P{=}SR$ where R is also a demonic channel. [10] We write $S{\sqsubseteq}P$ for that. □

The similarity between the three models is striking: in each case refinement is post-multiplication by a matrix of the same kind.

Demonic (secure-) refinement is reflexive and transitive but, as we observe in the example below, and as in the probabilistic case §2, the relation is not anti-symmetric: so far, we have only a pre-order.

Each column of the refinement matrix R makes a cell in P by taking the union of the S-cells that have *true* in that column. With that insight, we can rephrase Def. 3 as

$$\begin{array}{l} S \text{ is (demonic/secure) refined by } P \text{ iff for every cell of } P \\ \text{there is a set of cells of } S \text{ of which it is the union.} \end{array} \tag{1}$$

Put still another way, every cell of the more-refined P must be "justified" as the union of some set of cells in the less-refined S. An example of anti-symmetry's failure is then that $X = \{\ \{i_0\},\ \{i_1\},\ \{i_0,i_1\},\ \{i_0,i_1,i_2\}\ \}$ and $Y = \{\ \{i_0\},\ \{i_1\},\ \{i_0,i_1,i_2\}\ \}$ refine each other: to refine X to Y ignore the cell $\{i_0,i_1\}$ in X; to refine Y back to X merge the $\{i_0\}$ and $\{i_1\}$ in Y to replace the $\{i_0,i_1\}$ in X. Using 1 for *true* and 0 for *false*, with matrices we would have for X to Y the refinement

$$\begin{array}{cccccc} & X & R & & Y & \\ \hline \begin{array}{c} i_0 \\ i_1 \\ i_2 \end{array} & \begin{pmatrix} 1\,0\,1\,1 \\ 0\,1\,1\,1 \\ 0\,0\,0\,1 \end{pmatrix} & \begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \\ 0\,0\,1 \\ 0\,0\,1 \end{pmatrix} & = & \begin{pmatrix} 1\,0\,1 \\ 0\,1\,1 \\ 0\,0\,1 \end{pmatrix} & , \end{array}$$

where for example the third column of R shows that X's cells $\{i_0,i_1\}$ and $\{i_0,i_1,i_2\}$ are merged to a single cell $\{i_0,i_1,i_2\}$, and so $\{i_0,i_1\}$ is "lost". For

[9] We continue to call them "cells", as for partitions, in spite of the possible overlaps.
[10] We write $\mathcal{X}{\twoheadrightarrow}\mathcal{Y}$ for matrices (of any element-type) with $\mathcal{X}$-indexed rows and $\mathcal{Y}$-indexed columns. For deterministic matrices $\mathcal{I}{\twoheadrightarrow}\mathcal{O}$ is isomorphically functions $\mathcal{I}{\rightarrow}\mathcal{O}$.

the other direction Y to X we would have the matrices

$$\begin{array}{cccccc} & Y & R' & & X & \\ \hline \begin{array}{c} i_0 \\ i_1 \\ i_2 \end{array} & \begin{pmatrix} 1\,0\,1 \\ 0\,1\,1 \\ 0\,0\,1 \end{pmatrix} & \begin{pmatrix} 1\,0\,1\,0 \\ 0\,1\,1\,0 \\ 0\,0\,0\,1 \end{pmatrix} & = & \begin{pmatrix} 1\,0\,1\,1 \\ 0\,1\,1\,1 \\ 0\,0\,0\,1 \end{pmatrix} & , \end{array}$$

where the third column of R' "creates" $\{i_0, i_1\}$ from $\{i_0\}$ and $\{i_1\}$.

We achieve anti-symmetry via the usual closure construction.

Definition 4. *Union-closure for anti-symmetry* —— Say that a subset of $\mathbb{P}\mathcal{I}$ is *union closed* just when the union of each of its subsets is also an element of it. Define the *union closure* of some subset X of $\mathbb{P}\mathcal{I}$ to be the smallest union-closed subset of $\mathbb{P}\mathcal{I}$ that contains X, well defined because $\mathbb{P}\mathcal{I}$ is union-closed, and any intersection of union-closed sets is again union-closed. Write $X^{\cup}$ for the union-closure of X. □

Note that all union-closed subsets of $\mathbb{P}\mathcal{I}$ contain $\emptyset$, and so are non-empty. [11]

Lemma 1. *Anti-symmetry on union-closed sets* —— Take refinement $(\sqsubseteq)$ as in Def. 3. If $X, Y{:}\,\mathbb{P}\mathcal{I}$ are union-closed, with both $X \sqsubseteq Y$ and $Y \sqsubseteq X$, then in fact $X{=}Y$.

Proof Any element of Y must be the union of some subset of X and hence an element of $X^{\cup}$, which latter equals X again, because of its union-closure. □

Definition 5. *Demonic-refinement domain for information hiding*

Let $\mathbb{U}\mathcal{I}$ be the union-closed subsets of $\mathbb{P}\mathcal{I}$ that cover $\mathcal{I}$: it is the abstract model for demonic information-hiding. The refinement relation $(\sqsubseteq)$ is as defined above (Def. 3) for $\mathbb{P}\mathcal{I}$; but on $\mathbb{U}\mathcal{I}$ it is (also) antisymmetric, thus a partial order. □

Note that reflexivity and transitivity of $(\sqsubseteq)$ on $\mathbb{U}\mathcal{I}$ are inherited, since $\mathbb{U}\mathcal{I} \subseteq \mathbb{P}\mathcal{I}$.

Lemma 2. $\mathbb{U}\mathcal{I}$ *is a lattice* —— On $\mathbb{U}\mathcal{I}$ the refinement relation (Def. 3) is simply $(\supseteq)$. Thus for $X, Y{:}\,\mathbb{U}\mathcal{I}$, both therefore union-closed, their join $X \sqcup Y$ is simply $X \cap Y$, because it is union-closed as well and $(\supseteq)$ is a lattice. Their meet however needs explicit union closure: we define $X \sqcap Y$ to be $(X \cup Y)^{\cup}$.

Proof Omitted. □

3.2 Spies in action: an example of demonic nondeterminism

Recall the channels from Fig. 1. We can see that the union-closure of C^1 from Fig. 1(a) is $\{\, \emptyset, \mathsf{AE}, \mathsf{BW}, \mathsf{AEBW} \,\}$, where we write AE for $\{\mathsf{A}, \mathsf{E}\}$ etc. The union-closure of C^2 is $\{\, \emptyset, \mathsf{W}, \mathsf{ABE}, \mathsf{AEBW} \,\}$. Therefore from Lem. 2 the join $C^1 \sqcup C^2$ is $\{\, \emptyset, \mathsf{AEBW} \,\}$ as in Fig. 3, and the meet $C^1 \sqcap C^2$ is

$$\{\, \underline{\emptyset}, \mathsf{W}, \mathsf{AE}, \mathsf{BW}, \mathsf{ABE}, \underline{\mathsf{AEW}}, \mathsf{AEBW} \,\}\ , \tag{2}$$

[11] For any subset I of $\mathcal{I}$ we have $\emptyset \subseteq I$ and so $\emptyset = \cup\emptyset \in I^{\cup}$ also.

where the underlined $\underline{\emptyset}$ and $\underline{\mathsf{AEW}}$ have been added by union-closure (Lem. 2). We note however that (2) is *not* simply the union-closure of the meet Fig. 2(b) in the deterministic lattice: that would instead be $\{\mathsf{B},\mathsf{W},\mathsf{AE}\}^{\cup}$, that is

$$\{\ \underline{\emptyset}, \mathsf{B}, \mathsf{W}, \mathsf{AE}, \underline{\mathsf{BW}}, \underline{\mathsf{ABE}}, \underline{\mathsf{AEW}}, \underline{\mathsf{ABEW}}\ \}\ . \tag{3}$$

In fact in $\mathbb{U}\mathcal{I}$ we have $(3)\sqsubset(2)$ by discarding $\{\mathsf{B}\}$ from the former.

Thus in this case $\mathbb{U}\mathcal{I}$ admits a more-refined, that is a more *secure* meet (2) than the (3) admitted by $\mathbb{E}\mathcal{I}$; that is because (2) describes behaviour that no deterministic channel can realise, as we now discuss.

Suppose that $C^{1,2}$ are real spies, named *Ms. Vowel* and *Mrs. Early*, and our adversary M sends them into the field to discover the value of our hidden letter i. The mission however is so dangerous that she knows that only one of the spies will return: she just don't know beforehand which it will be. That is the nondeterminism. How do we describe this situation in $\mathbb{U}\mathcal{I}$?

In $\mathbb{U}\mathcal{I}$ this mission is in fact $C^1\sqcap C^2$, as in (2) and, as we remarked above, it is a strict refinement of the deterministic (3) where both spies return. The following lemma shows that (2) cannot be deterministic.

Lemma 3. *Characterisation of determinism within* $\mathbb{U}\mathcal{I}$ ___ For input space $\mathcal{I}$, the (union-closures of the) deterministic subset $\mathbb{E}\mathcal{I}$ of its demonic channels $\mathbb{U}\mathcal{I}$ comprise exactly those that are complement-closed. That is, any X in $\mathbb{U}\mathcal{I}$ is in fact $Y^{\cup}$ for some Y in $\mathbb{E}\mathcal{I}$ iff X is intersection- and complement-closed.

Proof "Only if" is trivial. If X in $\mathbb{U}\mathcal{I}$ is complement-closed, then it is also intersection-closed. For each i in $\mathcal{I}$ let X_i be the intersection of all elements (subsets of $\mathcal{I}$) of X that contain i. By intersection-closure of X each X_i is itself in X: in fact it is the smallest element of X that contains i.

Now for any two $i{\neq}i'$ we have that X_i and $X_{i'}$ are either equal or disjoint: if they had a proper overlap then either X_i or $X_{i'}$, or both, could be made smaller.

The sets X_i are the cells of the partition of which X is the union-closure: they are pairwise disjoint, non-empty, and cover $\mathcal{I}$. □

Lem. 3 shows that (2) cannot be deterministic, because it can reveal BW if Ms. Vowel returns (and says CONS); and it can also reveal ABW if Mrs. Early returns (saying EARLY). But this mission can never reveal B, that is the intersection $\mathsf{BW}\cap\mathsf{ABW}$, since for that both spies would have to return.

Now we consider an intriguing further possibility, where the spies report by radio instead of in person, using Booleans agreed beforehand (a one-time pad): for Ms. Vowel "*true*" encodes VOWEL etc. On this even more dangerous mission M knows that both spies will be captured, but she knows also that exactly one will send a report back to her by radio, either *true* or *false*. But she won't know which spy it was. Here the demonic channel is

$$\{\emptyset, \mathsf{BW}, \mathsf{ABE}, \mathcal{I}\} \tag{4}$$

which, by Lem. 3 again, is also properly demonic. This use of encoding, we should remark, underscores our abstraction from output values: from our point of view "Ms. Consonant" would be exactly the same spy as Ms. Vowel, and Mrs. Late would have the same utility as Mrs. Early.

3.3 Testing, soundness and completeness

The methodological concerns of §1.3 apply to demonic channels too: if we suspect that $S{\not\sqsubseteq}K$, how can we prove the refinement's failure in court?

Our earlier technique, for testing deterministic channels, does not work for demonic channels. † Let S be $\{\ \emptyset,\ \{i_0,i_1\},\ \{i_2\},\ \{i_0,i_1,i_2\}\ \}$ and K, not a refinement, be $\{\ \emptyset,\ \{i_0,i_1\},\ \{i_1,i_2\},\ \{i_0,i_1,i_2\}\ \}$. We know that $S{\not\sqsubseteq}K$ because $\{i_1,i_2\}$ in K is not the union of any cells in S. But no deterministic test ι in the style of §1.3 shows $S{\not\sqsubseteq}K$, because every cell of K is a superset of some cell of S. Thus deterministic tests are too weak, not complete for demonic channels. Strangely, every cell of K's being a superset of come cell of S, in a sense more demonic, is still *not* sufficient for refinement. [12]

In this section we synthesise a complete test-suite for demonic channels.

By definition we have $S{\not\sqsubseteq}K$ just when there is some cell κ in K that is not the union of any set of cells $\sigma_{1,\cdots,N}$ drawn from S — which, in turn, is just when there is further some single element i of $\mathcal{I}$ such that every i-containing cell σ of S is *not* a subset of κ. That is we have $S{\not\sqsubseteq}K$ just when

$$\begin{array}{l}\text{there are } i,\kappa \text{ with } i{\in}\kappa{\in}K \text{ such that for every } \sigma \text{ in } S\\ \text{we have } i{\in}\sigma \Rightarrow \sigma{\not\subseteq}\kappa\ .\end{array} \qquad (5)$$

Our preliminary definition of the "suite" of demonic tests is therefore that they are pairs (i,ι) with $i{\in}\iota{\subseteq}\mathcal{I}$. A demonic channel X passes such a test just when every cell χ in X with $i{\in}\chi$ satisfies $\chi{\not\subseteq}\iota$. [13]

For *soundness* of the (preliminary) test suite, argue the contrapositive by assuming that we have $S{\sqsubseteq}P$ and a test (i,ι) that P fails, so that there is some cell π in P with $i{\in}\pi{\subseteq}\iota$. But $\pi=\cup_n\sigma_n$ for some $\sigma_{1,\cdots,N}$, and so $i\in(\cup_n\sigma_n)\subseteq\iota$ whence, for some n, we have $i\in\sigma_n\subseteq\iota$ with $\sigma_n\in S$. That is, there is a cell σ_n of S that fails the test, and so S fails as a whole.

For *completeness* of the test suite, suppose $S{\not\sqsubseteq}K$ and appeal to (5) above to choose i,κ; then set $\iota{:=}\,\kappa$. The test (i,ι) itself is passed by S, by (5); but it is failed by K because we do not have $i{\in}\iota \Rightarrow \iota{\not\subseteq}\iota$ — the antecedent is true but the consequent is trivially false. For example the test that shows

$$\{\ \emptyset,\ \{i_0,i_1\},\ \{i_2\},\ \{i_0,i_1,i_2\}\ \} \quad \not\sqsubseteq \quad \{\ \emptyset,\ \{i_0,i_1\},\ \{i_1,i_2\},\ \{i_0,i_1,i_2\}\ \}\ ,$$

the example from (†) above, is $(i_1,\{i_1,i_2\})$ — the cells σ on the left that satisfy $i_1{\in}\sigma$ are $\{i_0,i_1\}$ and $\{i_0,i_1,i_2\}$ and, for both, we have $\sigma{\not\subseteq}\{i_1,i_2\}$. The cell $\kappa{:=}\,\{i_1,i_2\}$ on the right however satisfies $i_1{\in}\kappa$ but not of course $\kappa{\not\subseteq}\{i_1,i_2\}$.

For our preferred definition of demonic testing we reformulate the above in terms of two subsets of $\mathcal{I}$, rather than an element i and a subset ι, because that will be more convenient for source-level reasoning over programs. [14]

[12] They are trivially sound, however, since weakening a test suite trivially preserves its soundness: with fewer tests, there will be fewer failures.

[13] In fact $i{\in}\iota$ is not necessary, since a pair (i,ι) with $i{\notin}\iota$ would be a test passed by every cell, vacuously sound for all channels. Allowing it would make no difference.

[14] Subsets of $\mathcal{I}$, rather than individual elements, are more easily turned into predicates for source-level reasoning over a state space of typed variables: if you add another variable, a subset remains a subset but a point is no longer a point.

Definition 6. *Tests for demonic refinement* ___ A test for demonic refinement over space $\mathcal{I}$ is a pair (α,β) of subsets of $\mathcal{I}$. A demonic channel X passes the test (α,β) just when all its cells pass the test; a cell χ of X passes the test just when $\chi{\subseteq}\alpha \Rightarrow \chi{\subseteq}\beta$. $\square$

The top of the $\mathbb{U}\mathcal{I}$ lattice is the reveal-nothing channel $\{\emptyset,\mathcal{I}\}$, and it passes every non-trivial test; the bottom of the lattice is the reveal-everything channel $\mathbb{P}\mathcal{I}$ which fails them all. [15]

Lemma 4. *Equivalence of testing suites* ___ The test suite of Def. 6 is equivalent in power to the preliminary test suite (i,ι) discussed at (5).
Proof We show that $S{\not\sqsubseteq}K$ can be established by an (α,β)-test iff it can be established by an (i,ι)-test.

if — Any (i,ι)-test can be expressed as an (α,β)-test by setting $\alpha{:=}\,\iota$ and $\beta{:=}\,(\mathcal{I}{-}\{i\})$. To see that, let χ be an arbitrary cell and reason

$$\begin{array}{lll}
 & i{\in}\chi \Rightarrow \chi{\not\subseteq}\iota & \\
\text{iff} & \chi{\not\subseteq}(\mathcal{I}{-}\{i\}) \Rightarrow \chi{\not\subseteq}\iota & \\
\text{iff} & \chi{\subseteq}\iota \Rightarrow \chi{\subseteq}(\mathcal{I}{-}\{i\}) & \\
\text{iff} & \chi{\subseteq}\alpha \Rightarrow \chi{\subseteq}\beta\ . & \text{“set } \alpha,\beta{:=}\,\iota,(\mathcal{I}{-}\{i\})\text{”}
\end{array}$$

Thus (α,β)-tests are at least as discriminating as (i,ι)-tests.

only if — If $S{\not\sqsubseteq}K$ is established by (α,β), then for all cells σ in S we have $\sigma{\subseteq}\alpha \Rightarrow \sigma{\subseteq}\beta$; and for some cell κ in K we have $\kappa{\subseteq}\alpha \wedge \kappa{\not\subseteq}\beta$. Now reason

$$\begin{array}{lll}
 & \kappa{\subseteq}\alpha \wedge \kappa{\not\subseteq}\beta & \\
\text{iff} & \kappa{\subseteq}\alpha \wedge i{\in}\kappa & \text{“for some } i{\notin}\beta\text{”} \\
\text{hence} & \kappa \text{ fails test } (i,\alpha)\ , &
\end{array}$$

and for a contradiction

$$\begin{array}{lll}
\text{if} & \sigma \text{ fails test } (i,\alpha) & \text{“for the same } i{\notin}\beta \text{ as above”} \\
\text{then} & i{\in}\sigma \wedge \sigma{\subseteq}\alpha & \\
\text{hence} & i{\in}\sigma \wedge \sigma{\subseteq}\beta & \text{“assumption } \sigma \text{ passes test } (\alpha,\beta)\text{”} \\
\text{hence} & i{\in}\beta\ , &
\end{array}$$

which contradicts $i{\notin}\beta$, and so in fact σ cannot fail test (i,α).

Thus test (i,α) establishes $S{\not\sqsubseteq}K$, as required. $\square$

[15] Non-trivial tests make at least one distinction. Tests (α,β) are trivial when $\alpha{\subseteq}\beta$ (passed by every cell), and when α,β are disjoint (passed only by cell $\emptyset$.) In general (α,β) is equivalent to $(\alpha,\alpha{\cap}\beta)$.
Also for example (α',β') is weaker than (α,β) when $\alpha'{\subseteq}\alpha$ and $\beta{\subseteq}\beta'$. Compare Footnote 22 below.

Although $\mathbb{U}\mathcal{I}$ is restricted to union-closed subsets of $\mathcal{I}$, we can give an "abridged" representation of demonic channels in which union-closure is taken implicitly. In abridged form the non-refinement example from (†) becomes

$$\{\ \{i_0, i_1\},\ \{i_2\}\ \} \quad \not\sqsubseteq \quad \{\ \{i_0, i_1\},\ \{i_1, i_2\}\ \}\ ,$$

and the (α, β)-test for this non-refinement is $(\{i_1, i_2\}, \{i_0, i_2\})$. In fact we have

Lemma 5. *Testing abridged representations* ——— For any subset X of $\mathbb{P}\mathcal{I}$ and subsets α, β of $\mathcal{I}$, we have that X passes the test (α, β) iff the channel $X^{\cup}$ passes that same (α, β).
Proof If $X^{\cup}$ passes the test then so does X, because $X{\subseteq}X^{\cup}$.

If $X^{\cup}$ fails the test (α, β) then for some $\chi_{1,\cdots,N}$ in X we have $\cup(\chi_{1,\cdots,N}){\subseteq}\alpha$ but $\cup(\chi_{1,\cdots,N}){\not\subseteq}\beta$. From the latter we have $\chi_n{\not\subseteq}\beta$ for some n; but from the former we still have $\chi_n{\subseteq}\alpha$ for that n. Because that χ_n from X fails the test, so does X itself. □

From here on, we will use abridged representations if convenient. In fact, among abridged representations of a channel there is a smallest one where no cell is the union of any other cells (except itself). We call that the "reduced" representation of the channel, and note that all deterministic channels $\mathbb{E}\mathcal{I}$ are reduced.

Definition 7. *Reduced demonic channels* ——— A subset X of $\mathbb{P}\mathcal{I}$ is a *reduced* channel just when $\cup X = \mathcal{I}$ and no cell χ in X is the union $\cup\chi_{1,\cdots,N}$ of any other cells in X except trivially $\cup\{\chi\}$. Note that $\emptyset$ is excluded from an abridged representation, since it is $\cup\{\}$ (as well as $\cup\{\emptyset\}$.) □

We say that a reduced Y with $X{=}Y^{\cup}$ is a *reduction* of X.

Lemma 6. *Uniqueness of reductions* ——— Any demonic channel X in $\mathbb{U}\mathcal{I}$ has a unique reduction, a unique reduced channel Y in $\mathbb{P}\mathcal{I}$ such that $X{=}Y^{\cup}$.
Proof Existence of a reduction of X is trivial: keep removing superfluous cells in X until no more are superfluous.

For uniqueness we argue from Lem. 5 and the soundness of testing that two reductions Y, Y' of the same X must satisfy $Y{\sqsubseteq}Y'$ and $Y'{\sqsubseteq}Y$, so that any cell γ of Y is expressible as a union $\cup\gamma'_{1,\cdots,N}$ of cells γ'_n from Y'.

In turn, each of those γ'_n's must be a union of cells $\cup\gamma_{n,(1,\cdots,M)}$ back in Y, so that $\gamma = \cup\gamma_{(1,\cdots,N),(1,\cdots,M)}$.

Because Y is reduced, each $\gamma_{(1,\cdots,N),(1,\cdots,M)}$ must be just γ itself. Thus γ is in Y'. □

3.4 Justifying refinement's definition

The tests of Def. 6 justify a refinement failure $S{\not\sqsubseteq}K$ by guaranteeing that there is a test that S passes but K fails. The utility of a discriminating test is that, if you can find it, it proves the failure with a single witness. But the tests (α, β) are hardly an obvious, intuitive choice themselves.

To justify refinement's definition to both client and vendor, we appeal to a more primitive notion of correctness that we take as self-evidently desirable for security (of demonic channels): that if K can reveal its input is some i *exactly* but S never can, then K cannot be a refinement of S.

Definition 8. *Primitive refinement of channels* ———— We say that S is *primitively refined* by P just when there is no singleton cell $\{i\}$ in P that is not also in S. We write it $S{\preccurlyeq}P$. $\Box$

Put more simply, Def. 8 says that $S{\preccurlyeq}P$ unless there is a particular i that P can reveal but S cannot. "I might not know any theory; but I know that if S guarantees never to leak my password, then P can't either." [16]

It's the simplicity of ($\not\preccurlyeq$), in everyday terms, that is its justification. But it is however too simple for general use: Def. 8 does not justify ($\sqsubseteq$) directly. If S leaks the last character of a password, but K leaks the last two characters, then probably $S{\not\sqsubseteq}K$ — but we will still have $S{\preccurlyeq}K$ because neither leaks the password exactly.

Therefore to justify ($\sqsubseteq$) using Def. 8 we must do more: for that, we recognise that channels will probably not be used alone: larger channels can be made from a collection of smaller ones. In particular, we define

Definition 9. *Channel composition* The composition of two channels $C^{1,2}$ over the same input $\mathcal{I}$ but outputs $\mathcal{O}^{1,2}$ respectively a new channel of type $\mathcal{I}{\to}(\mathcal{O}^1{\times}\mathcal{O}^2)$ defined

$$(C^1\|C^2).i \quad := \quad (C^1.i, C^2.i) \ . \qquad \Box$$

Thus an adversary with access to two channels $C^{1,2}$ acting on the same input can be considered to be using a single channel $C^1\|C^2$: she observes its composite output (o^1, o^2) where $o^{1,2}{:=}C^{1,2}.i$ respectively.

We now give two desirable principles that should apply to ($\sqsubseteq$) in general: [17]

robustness If $S{\sqsubseteq}P$ then we should have primitive refinement even in the context of an arbitrary (other) channel C, that is $(S\|C) \preccurlyeq (P\|C)$.

necessity If $S{\not\sqsubseteq}P$ then for there must be some (other) channel C that justifies the failure, i.e. such that $(S\|C) \not\preccurlyeq (P\|C)$.

From the two principles above we can derive two others:

safety If $S{\sqsubseteq}P$ then $S{\preccurlyeq}P$, from applying *robustness* with the identity context.

monotonicity If $S{\sqsubseteq}P$ then $(S\|C){\sqsubseteq}(P\|C)$ for any (other) channel C — for, if not, by *necessity* there would be (still another) channel D such that $(S\|C)\|D \not\preccurlyeq (P\|C)\|D$, that is by associativity $S\|(C\|D) \not\preccurlyeq P\|(C\|D)$; and that, by robustness wrt. channel $C\|D$, implies $S{\not\sqsubseteq}P$.

[16] Just to be clear: a security breach releasing some large number N of passwords usually means in our terms that there are N singleton cells, not that there is just one cell with N passwords in it. The former means that each of N people has his password leaked exactly. The latter means instead that someone's password is leearned to be one of those N.

[17] Together they are an equivalence because $S{\sqsubseteq}P$ iff $(S\|C) \preccurlyeq (P\|C)$ for all C.

We note that the basic principles rest on two informal notions: that $(\not\preccurlyeq)$ reasonably captures "is clearly broken" in the sense a layman might understand it, and that $(\|C)$ describes "contexts" in which laymen would expect our channels reasonably to be able to operate. In particular, robustness emphasises that checking channels' security properties individually is not enough: two adversaries could have one channel each and, if they combined their results, they would in fact be acting like a single adversary using the channels' composition, probably a more powerful attack than is possible with either channel alone.

Once those notions $(\preccurlyeq)$ and $(\|C)$ are fixed, robustness and necessity determine refinement $(\sqsubseteq)$ uniquely. That is, justification of $(\preccurlyeq)$ and $(\|C)$ and robustness and necessity are collectively a justification of $(\sqsubseteq)$ and, further, it is the only relation that can be justified that way. This process is called *compositional closure*, that $(\sqsubseteq)$ is the compositional closure under $(\|)$ of $(\preccurlyeq)$.

The derived principles have direct significance for everyday use when a system $C_1\|\cdots\|C_N$ might be composed of many subsystems C_n: safety says that if a vendor establishes $S{\sqsubseteq}P$ through his software-development safe practices then, because (as well) he has established $S{\preccurlyeq}P$, his client will be happy; and monotonicity says that the vendor can use stepwise refinement [19] on his C_n's separately to modularise his software-development process that ultimately produces the whole system $C_1\|\cdots\|C_N$. We now have

Theorem 1. *Refinement is justified* ——— Def. 5 of refinement satisfies robustness and necessity wrt. Def. 8 (primitive refinement) and Def. 9 (composition).

Proof

Robustness

Assume that $S{\sqsubseteq}P$ but suppose for a contradiction that $S\|C \not\preccurlyeq P\|C$. In that case there must be π, γ from P, C respectively and input i such that the intersection $\pi{\cap}\gamma$ is $\{i\}$ for some i in $\mathcal{I}$, indicating than when the observation of $P\|C$ is (π,γ) an adversary would know that the input was i exactly, and
† furthermore that that does not occur with any σ from S. Now because $S{\sqsubseteq}P$ we have $\pi = \cup\sigma_{1,\cdots,N}$ for some $\sigma_{1,\cdots,N}$ each in S, so that

$$(\sigma_1{\cap}\gamma) \cup \cdots \cup (\sigma_N{\cap}\gamma) \quad = \quad \{i\} \quad \text{also,}$$

and so for at least one n we must have $\sigma_n{\cap}\gamma = \{i\}$, contradicting "furthermore" at (†) above. Thus $S\|C \preccurlyeq P\|C$ as desired.

Necessity

If $S{\not\sqsubseteq}K$ then by §3.3 (completeness and Def. 6) there is an (α,β) test that S passes but K fails. Choose therefore a cell κ in K such that $\kappa{\subseteq}\alpha$ but $\kappa{\not\subseteq}\beta$,
% and choose an element k in $\kappa{-}\beta$.

Define channel $C{:}\,\mathcal{I}{\rightarrow}Bool$ so that

$$C.i \quad := \quad i{=}k \;\vee\; i{\notin}\alpha \;, \tag{6}$$

and form channel $K \| C$ which for input k can give [18] output $(\kappa, true)$. In that case the adversary reasons

$$\begin{array}{lll}
 & i{\in}\kappa \ \wedge \ (i{=}k \vee i{\notin}\alpha) & \\
\text{implies} & i{\in}\alpha \ \wedge \ (i{=}k \vee i{\notin}\alpha) & \text{“}\kappa{\subseteq}\alpha\text{”} \\
\text{hence} & i{=}k \ , &
\end{array}$$

so that she deduces that i is k exactly.

Now we show that $S \| C$ can never reveal that $i{=}k$ exactly. If $S \| C$ is given (the same) input k then it will produce output $(\sigma, true)$ for some σ. Now assume for a contradiction that the adversary can deduce that i is k exactly in this case also. Write $\overline{\alpha}$ for $\mathcal{I}{-}\alpha$ and reason

$$\begin{array}{lll}
 & \sigma \cap (\{k\} \cup \overline{\alpha}) \ = \ \{k\} & \text{“assumption for contradiction”} \ ^{19} \\
\text{implies} & \sigma \cap (\{k\} \cup \overline{\alpha}) \cap \overline{\alpha} \ = \ \{k\} \cap \overline{\alpha} & \text{“}(\cap\overline{\alpha})\text{ both sides”} \\
\text{implies} & \sigma \cap \overline{\alpha} \ = \ \emptyset & \text{“}k \in \alpha\text{”} \\
\text{iff} & \sigma \subseteq \alpha & \\
\text{implies} & \sigma \subseteq \beta & \text{“assumption that } S \text{ passes test } (\alpha, \beta)\text{”} \\
\text{implies} & k \in \beta \ , & \text{“}k \in \sigma\text{”}
\end{array}$$

which contradicts that k was chosen from $\kappa{-}\beta$ at (%) above.
So we conclude that if $S{\not\sqsubseteq}K$ then there is an input k and a channel C such that running $K \| C$ on k can reveal k exactly but $S \| C$ on k can never reveal k exactly, that is that $S \| C \not\preccurlyeq K \| C$.

$\square$

Corollary 1. *Refinement is sound and monotonic* ——— Def. 5 of refinement satisfies soundness and monotonicity wrt. Def. 8 (primitive refinement) and Def. 9 (composition).
Proof Immediate from Thm. 1. $\square$

4 "'Weakest pre-tests" and source-level reasoning

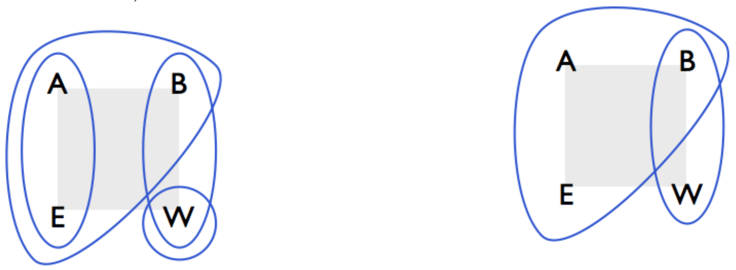
For eventual source-level reasoning, where e.g. leakage via channels is made a primitive imperative-programming statement, we can imagine asking what security guarantees we must have *before* a program runs in order to be sure that running the program has not leaked "too much" information *afterwards.*

Suppose that in our letters example it's especially important that the spies never learn that our i is exactly A, because A is information about an especially important person. For us the other people $\mathsf{B}, \mathsf{E}, \mathsf{W}$ are not so important.

[18] It's "can" rather than "must" because K is nondeterministic: it might not select cell κ for input k; but because $k{\in}\kappa$, it can.

[19] The left-hand side is the possibilities the observer deduces for the input i when she sees that $i{\in}\sigma$ and that $C.i{=}true$. The equality therefore says that she concludes the only possible input value is k.

Let our program (i.e. channel) be X with typical cell-names χ. To express "X never reveals that i is A" using a test in the style of Def. 6, we could write $\chi{\subseteq}\{\mathsf{A}\} \Rightarrow \chi{\subseteq}\emptyset$ for all $\chi{\in}X$. We can see by inspection from Fig. 2 that both the "one spy returns" channel and the "radio spies" channel pass that test (because all of their cells χ do).



(a) Only one spy returns. (b) A Boolean radio message is received.

Fig. 7. Ms. Vowel and Mrs. Early in action

So now we complicate things by imagining that, as a result of previous missions, M has some "a priori" knowledge about our i, knowledge that we would also like to express as a test. For example we could say that she knows *before* she sends Vowel and Early that i cannot be E, expressing that with the test
\$ $\chi{\subseteq}\mathcal{I} \Rightarrow \chi{\subseteq}\{\mathsf{ABW}\}$. Could she ever learn from her spies that actually $i{=}\mathsf{A}$?

The general "weakest pre-test" question for protecting A is [20]

> What security criterion must our i satisfy *before* the spies are sent in order to ensure that M cannot not learn $i{=}\mathsf{A}$ once the spies have reported?

Obviously the pre-test $i{\neq}\mathsf{A}$ would be strong enough — if you don't want A to be leaked, don't put it in your database. But can we do better?

The effect that M's a-priori knowledge, expressed as a cell μ say, has on her spies' output cells is simply that each cell χ becomes $\chi{\cap}\mu$ — she learns χ from the channel, and she knew μ already. Thus to convert our post-test $\chi{\subseteq}\{\mathsf{A}\}{\Rightarrow}\chi{\subseteq}\emptyset$ on χ to a pre-test on μ alone, we replace χ by $\chi{\cap}\mu$, to give

$$(\chi{\cap}\mu) \subseteq \{\mathsf{A}\} \quad \Rightarrow \quad (\chi{\cap}\mu) \subseteq \emptyset \;, \quad \text{and then} \tag{7}$$

instantiate that for every χ in the (abridged) channel Fig. 7(a) — we can do that because we know the channel's construction and so we know what χ's it can produce. If we take $\chi{=}\{\mathsf{W}\}$, we get $(\{\mathsf{W}\}{\cap}\mu) \subseteq \{\mathsf{A}\} \Rightarrow (\{\mathsf{W}\}{\cap}\mu) \subseteq \emptyset$, which is equivalent to $\mu \subseteq \mathsf{ABE} \Rightarrow \mu \subseteq \mathsf{ABE}$, that is just *true*. For all four cells it's

$$\begin{array}{llll}
\mu \subseteq \mathsf{ABE} \;\Rightarrow \mu \subseteq \mathsf{ABE} & \textit{true} & \text{when } \chi = \mathsf{W} & \text{(done just above)} \\
\mu \subseteq \mathsf{ABW} \Rightarrow \mu \subseteq \mathsf{BW} & \text{—} & \text{when } \chi = \mathsf{AE} & \\
\mu \subseteq \mathsf{AE} \;\;\Rightarrow \mu \subseteq \mathsf{AE} & \textit{true} & \text{when } \chi = \mathsf{BW} & \\
\mu \subseteq \mathsf{AW} \;\;\Rightarrow \mu \subseteq \mathsf{W} & \text{—} & \text{when } \chi = \mathsf{ABE}\,, &
\end{array} \tag{8}$$

[20] This is obviously by analogy with weakest preconditions [6].

where in each case we get a test again, but of "pre-cell" μ rather than "post-cell" χ, because $\chi\cap\mu \subseteq \iota$ can be written $\mu \subseteq \iota\cup\overline{\chi}$. Thus our overall pre-test for Fig. 7(a) and the post-test $\chi\subseteq\{\mathsf{A}\} \Rightarrow \chi\subseteq\emptyset$ is the conjunction

$$\mu \subseteq \mathsf{ABW} \Rightarrow \mu \subseteq \mathsf{BW} \quad \text{and} \quad \mu \subseteq \mathsf{AW} \Rightarrow \mu \subseteq \mathsf{W} \tag{9}$$

that we get by discarding the *true*'s from (8).

Thus a single post-test can generate a conjunction of pre-tests, which conjunctions we take therefore as our general form of test. [21] In this case however the first conjunct of (9) implies the second, and so we end up with only the first one. [22] To cast it back into everyday language, we rewrite (9) equivalently as $\mathsf{E}\notin\mu \Rightarrow \mathsf{A}\notin\mu$, that is that †

if M believes i is not E, she must also believe it's not A.

Under those conditions, her one-spy-returns attack will never reveal that $i{=}\mathsf{A}$.

In the case of the "radio spies" Fig. 7(b) we get only the second conjunct (because the case $\chi{=}\mathsf{AE}$ of (8) is missing), which as we have just seen is weaker than the first and so we can withstand "M's knowing more". That is, in Fig. 7(b) we are secure against M's knowing beforehand that $i{\neq}\mathsf{B}$ as at (\$) above; but in Fig. 7(a) we are not. That's not surprising, since Fig. 7(a) $\sqsubset$ Fig. 7(b) and therefore we expect to be less at risk from the radio spies.

For source-level reasoning we could e.g. write channels as primitive statements `leak c st` Φ`(c,i)` where Φ is a formula in state variables `i` and bound variable `c` is the emitted value: in state `i` the channel can emit any `c` satisfying Φ. As a special case we'd write `leak` ***Exp***`(i)` for the deterministic case, when Φ is `c=`***Exp***`(i)` for some expression ***Exp*** in `i`. A modality $\mathsf{K}\Psi(i)$ would express that the current cell χ satisfied $\chi \subseteq \{i{:}\mathcal{I}|\Psi(i)\}$, and our tests would then be of the form $(\forall c \bullet \mathsf{K}\Psi(i,c) \Rightarrow \mathsf{K}\Omega(i,c))$ where the universal quantifier would if necessary express wt-generated conjunctions (which distribute through subsequent wt's).

With all that, expressing our "weakest pre-test" approach at the source level (and making reference to variables implicit) would give in general

$$\begin{array}{ll} & \mathsf{wt}(\,\texttt{leak c st }\Phi(\texttt{c}),\ \mathsf{K}\Psi \Rightarrow \mathsf{K}\Omega\,) \\ = & (\forall c \bullet \mathsf{K}(\Phi \Rightarrow \Psi) \Rightarrow \mathsf{K}(\Phi \Rightarrow \Omega)) \ , \end{array}$$

and for the deterministic case $(\forall c \bullet \mathsf{K}(Exp{=}c \Rightarrow \Psi) \Rightarrow \mathsf{K}(Exp{=}c \Rightarrow \Omega))$.

The pre-test $\mathsf{E}\notin\mu \Rightarrow \mathsf{A}\notin\mu$ that we discovered at (†) above, to constrain M's prior knowledge, would be therefore be rendered at the source level as

$$\mathsf{K}(i{\neq}\mathsf{E}) \Rightarrow \mathsf{K}(i{\neq}\mathsf{A}) \qquad \text{If } M \text{ knows } i \text{ is not } \mathsf{E}, \text{ then she also knows it's not } \mathsf{A}.$$

[21] Starting again, from *conjunctions* of post-tests, will just generate conjunctions of conjunctions of pre-tests, so we do not have to expand our expressiveness any further. Furthermore, every member of $\mathbb{U}\mathcal{I}$ is characterised uniquely by a conjunction of such tests: every conjunction of tests "is" union-closed (easy); and for every union-closed set there is a conjunction of tests that only it satisfies (sneaky).

[22] In (9) here the (α',β') on the right is weaker than (α,β) on the left because we have $\alpha'{\subseteq}\alpha$ and $\alpha'{\cap}\beta \subseteq \beta'$. Compare Footnote 15 above.

Looking further ahead, we remark that for *updates* to the hidden state `i` the weakest pre-test is particularly simple, because updates leak nothing: if for example statement S is some assignment `i:=` Exp`(i)`, then the weakest pre-test is given by

$$\mathsf{wt}(\,S,\ \mathsf{K}\Psi \Rightarrow \mathsf{K}\Omega\,) \quad = \quad \mathsf{K}(\mathsf{wp}(S,\Psi)) \Rightarrow \mathsf{K}(\mathsf{wp}(S,\Omega))\ , \quad ^{23} \tag{10}$$

where wp is conventional weakest-precondition [6]; and this applies even when S is a demonic assignment (like a choice from a set). Non-leaking statements generate no pre-conjunctions. Conventional pre-and postconditions are embedded as "half tests" $\mathsf{K}(\mathit{true}) \Rightarrow \mathsf{K}\Omega$, equivalently just $\mathsf{K}\Omega$, and are respected by (10).

5 Conclusion

We have located a demonic model of information flow "in between" Landuaer and Redmond's deterministic model [8] and a more recent probabilistic model [9, 3]. Originally presented *ab initio* as "The Shadow" [15], it is now more clearly structured; and as a result its properties can be divided into those inherent in demonic choice, and those shared with other models of information flow.

The deterministic model is a restriction (not an abstraction) of the demonic model: they give the same level of detail, but the latter describes more situations than the former. For example collaboration of the Spies (§3.2) cannot be expressed at all in the deterministic mode. The demonic model is however an abstraction (not a restriction) of the probabilistic: they can describe the same systems, but the latter gives a more detailed (i.e. quantitative rather than only qualitative) description. For the Spies, we are abstracting from the probabilities that one or the other might return, and the prior probability on the secret letter $\mathsf{A}, \mathsf{B}, \mathsf{E}, \mathsf{W}$.

All three systems have the the same structural definition of secure refinement, particularly evident when we use the matrix formulation: one channel P is a secure refinement of another channel S just when P can be obtained via post-multiplication by a so-called refinement matrix. This is in fact channel cascade, if the refinement matrix is for that purpose considered to be a channel from S-observables to P-observables.

The deterministic- and the demonic systems are lattices wrt. the refinement order; but the probabilistic system is not [12]: it is however a partial order if properly quotiented.

All three systems have a complementary testing semantics, one that provides a witness to any refinement failure. All three systems can justify their refinement order by general principles, robustness and necessity (§3.4) whereby the refinement relation is reduced to a more primitive form that is accepted "by the layman". (In the probabilistic case, the reduction is to the more primitive *Bayes Vulnerability*, the probability of guessing the secret in one try [9, 5].)

[23] We assume here that S is everywhere terminating.

Finally, we mention that these systems show how the notion of security has become more sophisticated over the decades. Originally a system was said to be secure or insecure, an absolute black-or-white judgement, based on whether is suffered from "interference" or not [7]. Later it was realised that this criterion is too strong, since almost no useful system can be wholly interference-free: even a password-based login system releases information when a login attempt fails.

That led to the idea comparing two programs' information flow, particularly comparing a specification with an implementation: refinement holds just when the implementation cannot leak except when the specification can. In the probabilistic case, the comparison is even more sophisticated: the implementation must leak *no more than* the specification does.

Our aim is to enable this kind of refinement-based reasoning at the source-code level, based on "information-flow aware" assertions like those proposed in §4. From those it should be possible to construct an algebra of program transformations that preserve security- and functional characteristics during the program-development process in which specifications are manipulated to become implementations.

Finally, in the longer term we would like to add a fourth layer above the three mentioned here: one where probability, demonic choice and secrecy are handled all at once.

References

1. Mário S. Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. Additive and multiplicative notions of leakage, and their capacities. In *IEEE 27th CSF 2014*, pages 308–322. IEEE, 2014.
2. Mário S. Alvim, Konstantinos Chatzikokolakis, Annabelle McIver, Carroll Morgan, Catuscia Palamidessi, and Geoffrey Smith. Axioms for information leakage. In *Proc. CSF 2016*, pages 77–92, June 2016.
3. Mário S. Alvim, Kostas Chatzikokolakis, Catuscia Palamidessi, and Geoffrey Smith. Measuring information leakage using generalized gain functions. In *Proc. 25th IEEE (CSF 2012)*, pages 265–279, June 2012.
4. D. Blackwell. Comparison of experiments. In *Proc. Second Berkeley Symposium on Mathematical Statistics and Probability*, pages 93–102, 1951.
5. Nicolás E. Bordenabe and Geoffrey Smith. Correlated secrets in information flow. In *Proc. CSF 2016*, pages 93–104, June 2016.
6. E.W. Dijkstra. *A Discipline of Programming.* Prentice-Hall, 1976.
7. J.A. Goguen and J. Meseguer. Unwinding and inference control. In *Proc. IEEE Symp on Security and Privacy*, pages 75–86. IEEE Computer Society, 1984.
8. Jaisook Landauer and Timothy Redmond. A lattice of information. In *Proc. 6th IEEE CSFW'93*, pages 65–70, June 1993.
9. Annabelle McIver, Larissa Meinicke, and Carroll Morgan. Compositional closure for Bayes Risk in probabilistic noninterference. In *Proceedings of the 37th international colloquium conference on Automata, languages and programming: Part II*, volume 6199 of *ICALP'10*, pages 223–235, Berlin, Heidelberg, 2010. Springer.
10. Annabelle McIver, Larissa Meinicke, and Carroll Morgan. A Kantorovich-monadic powerdomain for information hiding, with probability and nondeterminism. In *Proc. LiCS 2012*, 2012.

11. Annabelle McIver, Larissa Meinicke, and Carroll Morgan. Hidden-Markov program algebra with iteration. *Mathematical Structures in Computer Science*, 2014.
12. Annabelle McIver, Carroll Morgan, Larissa Meinicke, Geoffrey Smith, and Barbara Espinoza. Abstract channels, gain functions and the information order. In *FCS 2013*, 2013.
http://prosecco.gforge.inria.fr/personal/bblanche/fcs13/fcs13proceedings.pdf.
13. Annabelle McIver, Carroll Morgan, and Tahiry Rabehaja. Abstract Hidden Markov Models: a monadic account of quantitative information flow. In *Proc. LiCS 2015*, 2015.
14. Annabelle McIver, Carroll Morgan, Geoffrey Smith, Barbara Espinoza, and Larissa Meinicke. Abstract channels and their robust information-leakage ordering. In Martín Abadi and Steve Kremer, editors, *POST 2014, ETAPS 2014, Proceedings*, volume 8414 of *Lecture Notes in Computer Science*, pages 83–102. Springer, 2014.
15. C.C. Morgan. *The Shadow Knows:* Refinement of ignorance in sequential programs. In T. Uustalu, editor, *Math Prog Construction*, volume 4014 of *Springer*, pages 359–78. Springer, 2006. Treats *Dining Cryptographers*.
16. C.C. Morgan. *The Shadow Knows:* Refinement of ignorance in sequential programs. *Science of Computer Programming*, 74(8):629–653, 2009. Treats *Oblivious Transfer*.
17. C.E. Shannon. A mathematical theory of communication. *Bell System Technical Journal*, 27:379–423, 623–656, 1948.
18. Geoffrey Smith. On the foundations of quantitative information flow. In Luca de Alfaro, editor, *Proc. FoSSaCS '09*, volume 5504 of *Lecture Notes in Computer Science*, pages 288–302, 2009.
19. N. Wirth. Program development by stepwise refinement. *Comm ACM*, 14(4):221–7, 1971.