Lecture 14

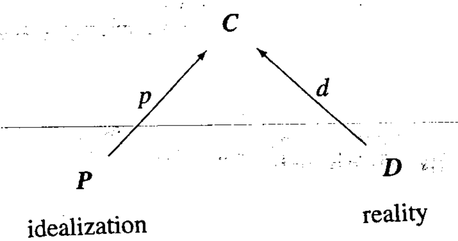
Representing Local Logics

We have now presented two mathematical models of information flow in distributed systems, information channels, and local logics. In this lecture we explore the relationship between these models.

14.1 Idealization

An old problem in the philosophy of mathematics and science has to do with the efficacy of mathematics in understanding the real world. This can be put as a question of information flow. How is it that science, with its use of abstract mathematical models, carries any information at all about the real world? We want to look at this question in terms of information channels as a way to motivate our discussion of the relation between local logics and binary information channels.

Suppose we have a binary channel C relating $\boldsymbol{P}$ and $\boldsymbol{D}$ as depicted below.



Here the classification $\boldsymbol{D}$ is a domain under scientific investigation, in all its real-world complexity, and $\boldsymbol{P}$ is some specific scientific idealization or model of the domain $\boldsymbol{D}$. (The "P" and "D" stand for "proximal" and "distal", respectively, as in Lecture 2.) The channel C models the idealization or modeling

process being used. A token c of $\boldsymbol{C}$ connects some particular situation $a = d(c)$ in $\boldsymbol{D}$ and an idealized token or model $b = p(c)$ in $\boldsymbol{P}$, often some mathematical object. This channel C is itself a model of the use of the idealized domain $\boldsymbol{P}$ to reason about the reality $\boldsymbol{D}$ in terms of moving the natural logic $\text{Log}(\boldsymbol{P})$ on $\boldsymbol{P}$ to $\boldsymbol{D}$ along this channel. This prompts the following definition.

Definition 14.1. Given a binary channel C from $\boldsymbol{P}$ to $\boldsymbol{D}$, the local logic on $\boldsymbol{D}$ *induced by* C is the logic

$$\text{Log}_C(\boldsymbol{D}) = d^{-1}[p[\text{Log}(\boldsymbol{P})]].$$

The concept of a local logic gives us the wherewithal to capture the structure of reasoning by means of idealizations. The logic $\text{Log}_C(\boldsymbol{D})$ on $\boldsymbol{D}$ is not guaranteed to be complete; that assurance is lost by taking the image under p. Neither is it guaranteed to be sound; that assurance is lost by taking the inverse image under d. The constraints of $\text{Log}_C(\boldsymbol{D})$ are those that come from the idealization process from some law that holds in the idealized domain. The normal tokens of $\text{Log}_C(\boldsymbol{D})$ are those real-world tokens that are "appropriately connected" to some idealized token. The better the scientific model is, the better this logic is, better in the sense of having fewer exceptional tokens and more constraints.

As a theory of idealization, scientific modeling, and the efficacy of applied mathematics this sketch is at best highly programmatic. Some further thoughts related to this sketch are presented in Lecture 20. Its purpose here is to introduce the definition of the logic induced by a binary channel. We can characterize this induced logic in the following way.

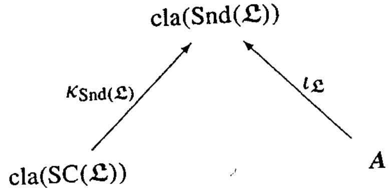
Proposition 14.2. *Let C be a binary channel from $\boldsymbol{P}$ to $\boldsymbol{D}$ as depicted above.*

1. *A partition $\langle\Gamma, \Delta\rangle$ of* $\text{typ}(\boldsymbol{D})$ *is consistent in* $\text{Log}_C(\boldsymbol{D})$ *if and only if $\langle p^{-1}[d[\Gamma]], p^{-1}[d[\Delta]]\rangle$ is the state description of some $b \in \text{tok}(\boldsymbol{P})$.*
2. *A token $a \in \text{tok}(\boldsymbol{D})$ is normal in $\text{Log}_C(\boldsymbol{D})$ if and only if it is connected to a token of $\boldsymbol{P}$.*

Proof. This characterization is obtained by applying the definitions of images and inverse images in Definitions 13.1 and 13.2. □

Example 14.3. Recall that for any logic $\mathfrak{L}$ on any classification $\boldsymbol{A}$, the sound completion $\text{SC}(\mathfrak{L})$ is the logic that is obtained by throwing away exceptional tokens and adding in idealized tokens for the consistent partitions that are not realized. This is clearly some kind of idealization process. Let us see that it fits the above picture.

Because $SC(\mathfrak{L})$ is sound and complete, it is a natural logic. In other words, if we let $\boldsymbol{P}$ be the classification of $SC(\mathfrak{L})$, then $SC(\mathfrak{L}) = Log(\boldsymbol{P})$. Recalling the inclusion infomorphisms $\iota_{\mathfrak{L}} : \mathfrak{L} \rightleftarrows Snd(\mathfrak{L})$ and $\kappa_{Snd(\mathfrak{L})} : SC(\mathfrak{L}) \rightleftarrows Snd(\mathfrak{L})$, we have the following channel $\mathcal{C}$:



With this channel, we have $\mathfrak{L} = Log_{\mathcal{C}}(\boldsymbol{A})$.

We can summarize this discussion as follows.

Theorem 14.4. *Every local logic $\mathfrak{L}$ on a classification $\boldsymbol{A}$ is of the form $Log_{\mathcal{C}}(\boldsymbol{A})$ for a binary channel $\mathcal{C}$ linking $\boldsymbol{A}$ to the classification of the sound completion of $\mathfrak{L}$.*

This shows that all local logics are induced by binary channels. The sound completion of $\mathfrak{L}$ is not well behaved with regard to infomorphisms, however. That is, logic infomorphisms do not give us the right kind of morphisms between their associated channels when one uses the sound completion as an idealization. There is a closely related idealization associated with the local logic $\mathfrak{L}$ that works out better.

Definition 14.5. Given a local logic $\mathfrak{L}$, the *idealization of* $\mathfrak{L}$ is the classification $Idl(\mathfrak{L})$ with the same types as $\mathfrak{L}$, whose tokens are the consistent partitions of $\mathfrak{L}$, and such that $\langle \Gamma, \Delta \rangle \vDash_{Idl(\mathfrak{L})} \alpha$ if and only if $\alpha \in \Gamma$ (equivalently, if and only if $\alpha \notin \Delta$).

We have seen this classification earlier in various guises.

Proposition 14.6. *For any logic $\mathfrak{L}$,*

1. $Idl(\mathfrak{L}) = Cla(th(\mathfrak{L}))$, *and*
2. $Idl(\mathfrak{L}) \cong Sep(cla(SC(\mathfrak{L})))$.

Proof. The first part is clear from the definitions of the two sides. For the second part, let $g : Idl(\mathfrak{L}) \rightleftarrows cla(SC(\mathfrak{L}))$ be the type-identical infomorphism such that

$g(a)$ is the state description of a in $cla(SC(\mathfrak{L}))$, which is a consistent partition (by the soundness of $SC(\mathfrak{L})$) and so is in $tok(Idl(\mathfrak{L}))$. This is clearly a bijection on tokens. It is easy to check that g is an infomorphism. □

Definition 14.7. Given a local logic $\mathfrak{L}$, the channel *representing* $\mathfrak{L}$ is the binary channel $Cha(\mathfrak{L}) = \langle h_{cla(\mathfrak{L})}, h_{Idl(\mathfrak{L})} \rangle$ from $cla(\mathfrak{L})$ to $Idl(\mathfrak{L})$ with core $cla(Snd(\mathfrak{L}))$ and type-identical infomorphisms given by

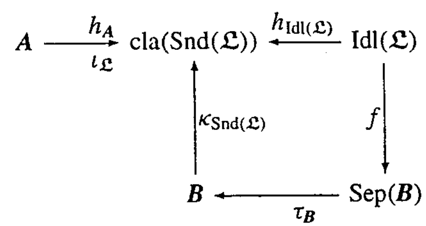
1. $h_{cla(\mathfrak{L})}(a) = a$, and
2. $h_{Idl(\mathfrak{L})}(a) = state_{\mathfrak{L}}(a)$

for each token $a \in tok(Snd(\mathfrak{L}))$.

Justification. These are indeed infomorphisms: $h_{cla(\mathfrak{L})}$ is just the inclusion infomorphism $\iota_{cla(Snd(\mathfrak{L}))}$ from Proposition 13.10, and $h_{Idl(\mathfrak{L})}$ is an infomorphism because the state description of every token of $Snd(\mathfrak{L})$ is consistent and so is a token of $Idl(\mathfrak{L})$, by the soundness of $Snd(\mathfrak{L})$. □

Theorem 14.8. *Every local logic $\mathfrak{L}$ on a classification $\boldsymbol{A}$ is the derived logic of its associated channel; that is, $Log_{Cha(\mathfrak{L})}(\boldsymbol{A}) = \mathfrak{L}$.*

Proof. Let $\boldsymbol{B} = cla(SC(\mathfrak{L}))$. By Proposition 14.6, there is an isomorphism $f : Idl(\mathfrak{L}) \rightleftarrows Sep(\boldsymbol{B})$. Thus we have the following diagram:



The diagram commutes, by Exercise 4.5, because all the infomorphisms are type identical, and so the image of $Log(Idl(\mathfrak{L}))$ under $h_{Idl(\mathfrak{L})}$ is the same as its image under $\kappa_{Snd(\mathfrak{L})} \tau_{\boldsymbol{B}} f$. Thus

$$h_{Idl(\mathfrak{L})}[Log(Idl(\mathfrak{L}))] = \kappa_{Snd(\mathfrak{L})}[\tau_{\boldsymbol{B}}[f[Log(Idl(\mathfrak{L}))]]]$$

by Proposition 13.13. Note that

1. $f[Log(Idl(\mathfrak{L}))] = Log(Sep(\boldsymbol{B}))$ because f is an isomorphism, and

2. $\tau_B[\mathrm{Log}(\mathrm{Sep}(B))] = \mathrm{Log}(B)$ because τ_B is surjective on tokens, by Proposition 13.12,

and so $h_{\mathrm{Idl}(\mathfrak{L})}[\mathrm{Log}(\mathrm{Idl}(\mathfrak{L}))] = \kappa_{\mathrm{Snd}(\mathfrak{L})}[\mathrm{Log}(B)]$. But then $\mathrm{Log}(\mathrm{Cha}(\mathfrak{L}))$ is $\iota_A^{-1}[\kappa_{\mathrm{Snd}(\mathfrak{L})}[\mathrm{Log}(B)]]$, which is just $\mathfrak{L}$. □

Corollary 14.9. *Every sound local logic $\mathfrak{L}$ is the image of its idealization $\mathrm{Idl}(\mathfrak{L})$ under the infomorphism $h_{\mathrm{Idl}(\mathfrak{L})}$.*

Proof. If $\mathfrak{L}$ is sound, then the other half of the channel $\mathrm{Cha}(\mathfrak{L})$ is the identity. □

14.2 Channel Infomorphisms

We have associated an information channel $\mathrm{Cha}(\mathfrak{L})$ with each local logic $\mathfrak{L}$. If this association is natural, there should be a correspondence between logic infomorphisms and channel infomorphisms. We have not defined the latter notion, though it is in some sense implicit in our notion of a refinement between channels. We define this notion now in order to show that logic infomorphisms do indeed naturally correspond to channel infomorphisms. The material in this section will not be needed in for the following lectures.

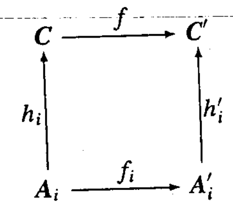
Definition 14.10. Given channels

$$\mathcal{C} = \{h_i : A_i \rightleftarrows C\}_{i \in I}$$

and

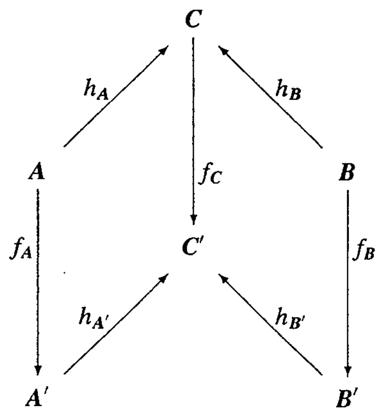
$$\mathcal{C}' = \{h_i' : A_i' \rightleftarrows C'\}_{i \in I}$$

on an index set I, a *channel infomorphism* $f : \mathcal{C} \rightleftarrows \mathcal{C}'$ consists of infomorphisms $f : C \rightleftarrows C'$ and $f_i : A_i \rightleftarrows A_i'$ such that for each $i \in I$ the following diagram commutes:



Thus a channel refinement is essentially a channel infomorphism $f : \mathcal{C} \rightleftarrows \mathcal{C}'$, where $A_i = A_i'$ and $f_i = 1_{A_i}$ for each $i \in I$.

Proposition 14.11. *Suppose we have a channel infomorphism $f : \mathcal{C} \rightleftarrows \mathcal{C}'$ between binary channels as depicted by the following diagram:*



Then f_A, considered as a map of logics, is a logic infomorphism of the induced logics $f_A : \mathrm{Log}_{\mathcal{C}}(A) \rightleftarrows \mathrm{Log}_{\mathcal{C}'}(A)$.

Proof. First we show that f_A on types is a regular theory interpretation. Suppose $\langle \Gamma', \Delta' \rangle$ is a consistent partition of $\mathrm{Log}_{\mathcal{C}'}(A')$. By Proposition 14.2, there is a $b' \in \mathrm{tok}(B'$ whose state description is $\langle h_{B'}^{-1}[h_{A'}[\Gamma]], h_{B'}^{-1}[h_{A'}[\Delta]] \rangle$. Then $f_B(b')$ has state description $\langle f_B^{-1}[h_{B'}^{-1}[h_{A'}[\Gamma]]], f_B^{-1}[h_{B'}^{-1}[h_{A'}[\Delta]]] \rangle$. From the diagram, we see that

$$f_B^{-1} h_{B'}^{-1} h_{A'} = h_B^{-1} f_C^{-1} h_{A'} = h_B^{-1} h_A f_A^{-1}.$$

Thus $f_B(b')$ has state description $\langle h_B^{-1}[h_A[f_A^{-1}[\Gamma']]], h_B^{-1}[h_A f_A^{-1}[\Delta']] \rangle$. Applying Proposition 14.2 again, $\langle f_A^{-1}[\Gamma'], f_A^{-1}[\Delta'] \rangle$ is consistent and so f_A is a theory interpretation, as required.

Let us check that if a' is a normal token of $\mathrm{Log}_{\mathcal{C}'}(A')$, then $f_A(a')$ is a normal token of $\mathrm{Log}_{\mathcal{C}}(A)$. By Proposition 14.2, if a' is a normal token of $\mathrm{Log}_{\mathcal{C}'}(A')$, then there is a connection c' of $\mathcal{C}'$ connecting a' to some token b' of B'. But then $f_C(c')$ is a connection of $\mathcal{C}$ connecting $f_A(a')$ to $f_B(b')$. Hence by Proposition 14.2, $f_A(a')$ is a normal token of $\mathrm{Log}_{\mathcal{C}}(A)$. □

The preceding result shows that channel infomorphisms give rise to logic infomorphisms. We now prove that every logic infomorphism can be obtained in this way when we restrict attention to channels that represent logics.

Proposition 14.12. *Given a logic infomorphism $f : \mathfrak{L} \rightleftarrows \mathfrak{L}'$, there are unique infomorphisms $f_{\mathrm{Snd}(\mathfrak{L})} : \mathrm{cla}(\mathrm{Snd}(\mathfrak{L})) \rightleftarrows \mathrm{cla}(\mathrm{Snd}(\mathfrak{L}'))$ and $f_{\mathrm{Idl}(\mathfrak{L})} : \mathrm{Idl}(\mathfrak{L}) \rightleftarrows$*

Idl($\mathfrak{L}'$) *such that the following diagram commutes:*

$$\begin{array}{ccccc}
\text{cla}(\mathfrak{L}) & \xrightarrow{h_{\text{cla}(\mathfrak{L})}} & \text{cla}(\text{Snd}(\mathfrak{L})) & \xleftarrow{h_{\text{Idl}(\mathfrak{L})}} & \text{Idl}(\mathfrak{L}) \\
\downarrow f & & \downarrow f_{\text{Snd}(\mathfrak{L})} & & \downarrow f_{\text{Idl}(\mathfrak{L})} \\
\text{cla}(\mathfrak{L}') & \xrightarrow{h_{\text{cla}(\mathfrak{L}')}} & \text{cla}(\text{Snd}(\mathfrak{L}')) & \xleftarrow{h_{\text{Idl}(\mathfrak{L})}} & \text{Idl}(\mathfrak{L}')
\end{array}$$

Proof. To satisfy the conditions of the proposition, $f_{\text{Snd}(\mathfrak{L})}$ and $f_{\text{Idl}(\mathfrak{L})}$ must satisfy the following:

1. $f_{\text{Snd}(\mathfrak{L})}(\alpha) = f(\alpha)$ for each $\alpha \in \text{typ}(\text{Snd}(\mathfrak{L}))$,
2. $f_{\text{Snd}(\mathfrak{L})}(b) = f(b)$ for each $b \in \text{tok}(\text{Snd}(\mathfrak{L}'))$,
3. $f_{\text{Idl}(\mathfrak{L})}(\alpha) = f(\alpha)$ for each $\alpha \in \text{typ}(\text{Idl}(\mathfrak{L}))$, and
4. $f_{\text{Idl}(\mathfrak{L})}(\langle \Gamma, \Delta \rangle) = \langle f^{-1}[\Gamma], f^{-1}[\Delta] \rangle$ for each $\langle \Gamma, \Delta \rangle \in \text{tok}(\text{Idl}(\mathfrak{L}'))$.

But these amount to a definition of $f_{\text{Snd}(\mathfrak{L})}$ and $f_{\text{Idl}(\mathfrak{L})}$, so ensuring the existence and uniqueness of these infomorphisms. □

Definition 14.13. Given a logic infomorphism $f : \mathfrak{L} \rightleftarrows \mathfrak{L}'$, let

$$\text{Cha}(f) : \text{Cha}(\mathfrak{L}) \rightleftarrows \text{Cha}(\mathfrak{L}')$$

be the channel infomorphism given by $f : \text{cla}(\mathfrak{L}) \rightleftarrows \text{cla}(\mathfrak{L}')$, $f_{\text{Snd}(\mathfrak{L})} : \text{cla}(\text{Snd}(\mathfrak{L})) \rightleftarrows \text{cla}(\text{Snd}\mathfrak{L}'))$, and $f_{\text{Idl}(\mathfrak{L})} : \text{Idl}(\mathfrak{L}) \rightleftarrows \text{Idl}(\mathfrak{L}')$.

Justification. This is a channel infomorphism by Proposition 14.12. □

Proposition 14.14. *For any logic infomorphism* $f : \mathfrak{L} \rightleftarrows \mathfrak{L}'$,

$$\text{Log}(\text{Cha}(f)) = f.$$

Proof. This is immediate from the definitions. □

We take these results to show that we have the "right" information-theoretic definition of morphism between channels, but we do not pursue this topic further here.

Exercises

14.1. Show that the identity function from the types of th($\mathfrak{L}$) to the types of Th(cla(Snd($\mathfrak{L}$))) is a theory interpretation and that the infomorphism $h_{\text{Idl}(\mathfrak{L})}$ in Definition 14.7 is its image under Cla.

14.2. (†) Call a binary channel $\mathcal{C}$ from $\boldsymbol{A}$ to $\boldsymbol{B}$ *special* if there is a function $f : \text{typ}(\boldsymbol{A}) \to \text{typ}(\boldsymbol{B})$ such that $h_{\boldsymbol{B}} f = h_{\boldsymbol{A}}$.

1. Show that for any special channel $\mathcal{C}$ and logic infomorphism $f : \mathfrak{L} \rightleftarrows \text{Log}_{\mathcal{C}}(\boldsymbol{A})$, there is a unique channel infomorphism $f^* : \text{Cha}(\mathfrak{L}) \rightleftarrows \mathcal{C}$ such that $\text{Log}(f^*) = f$.
2. Use (1) to show that Cha is left adjoint to the functor Log restricted to special channels.
3. What is the counit of this adjoint pair?

Lecture 15

Distributed Logics

The view of information put forward here associates information flow with distributed systems. Such a system $\mathcal{A}$, we recall, consists of an indexed family $\text{cla}(\mathcal{A}) = \{A_i\}_{i \in I}$ of classifications together with a set $\text{inf}(\mathcal{A})$ of infomorphisms, all of which have both a domain and a codomain in $\text{cla}(\mathcal{A})$. With any such a system we want to associate a systemwide logic $\text{Log}(\mathcal{A})$ on the sum $\sum_{i \in I} A_i$ of the classifications in the system. The constraints of $\text{Log}(\mathcal{A})$ should use the lawlike regularities represented by the system as a whole. The normal tokens of $\text{Log}(\mathcal{A})$ model those indexed families of tokens to which the constraints are guaranteed to apply, by virtue of the structure of the system.

If we consider a given component classification A_i of $\mathcal{A}$, there are at least two sensible logics on A_i that we might want to incorporate into $\text{Log}(\mathcal{A})$, the *a priori* logic $\text{AP}(A_i)$ and the natural logic $\text{Log}(A_i)$. The former assumes we are given no information about the constraints of A_i except for the trivial constraints. The latter assumes perfect information about the constraints of A_i. There is typically quite a difference. But really these are just two extremes in our ordering of sound local logics on A_i. After all, in dealing with a distributed system, we may have not just the component classifications and their informorphisms, but also local logics on the component classifications. We want the systemwide logic to incorporate these local logics. To this end, we generalize the notion of a distributed system to that of an information system.

15.1 Information Systems

Definition 15.1. An *information system* $\mathcal{L}$ consists of an indexed family $\log(\mathcal{L}) = \{\mathfrak{L}_i\}_{i \in I}$ of local logics together with a set $\text{inf}(\mathcal{L})$ of logic infomorphisms, all of which have both a domain and a codomain in $\log(\mathcal{L})$.

We associate with any information system $\mathcal{L}$ a systemwide logic $\text{Log}(\mathcal{L})$, one that is the "limit" of the logics of the system and the way they fit together.

Before defining this logic, it is worthwhile to remind ourselves that once we have a logic on a sum $\sum_{i \in I} A_i$, we get an associated logic on any smaller sum $\sum_{i \in I_0} A_i$, for $I_0 \subset I$, as in Example 13.8, by simply restricting the constraints to those of the smaller sum and taking as normal tokens those that are projections of normal tokens in the larger sum. We will often be interested in some such restricted logic of the system.

Definition 15.2. Let $\mathcal{L}$ be an information system, that is, an indexed family $\log(\mathcal{L}) = \{\mathfrak{L}_i\}_{i \in I}$ of local logics together with a set $\text{inf}(\mathcal{L})$ of logic infomorphisms, all of which have both a domain and a codomain in $\log(\mathcal{L})$. Let $A_i = \text{cla}(\mathfrak{L}_i)$, and let $\mathcal{A}$ be the associated distributed system with the same set of infomorphisms. Let $\mathcal{C} = \lim \mathcal{A}$ be the limit of this distributed system; write this channel as an indexed family $\{g_i : A_i \rightleftarrows C\}_{i \in I}$. There is a least logic $\mathfrak{L}$ on C such that each g_i is a logic infomorphism. Namely, $\mathfrak{L} = \bigsqcup_{i \in I} g_i[\mathfrak{L}_i]$. The logic $\text{Log}(\mathcal{L})$ has classification $A = \sum_{i \in I} A_i$. If we let $g = \sum_{i \in I} g_i$, where we think of these as classification infomorphisms, then $g : A \rightleftarrows C$ is an infomorphism. We define $\text{Log}(\mathcal{L}) = g^{-1}[L]$; that is, it is the largest logic on A that makes g a logic infomorphism. In summary, then, the *distributed logic* of the information system $\mathcal{L}$ is the local logic on $\sum_{i \in I} A_i$ given by

$$\text{Log}(\mathcal{L}) = \left(\sum_{i \in I} g_i\right)^{-1} \left[\bigsqcup_{i \in I} g_i[\mathfrak{L}_i]\right].$$

This definition gives us a way to state our basic proposal for an understanding of information flow in a distributed system.

Basic Proposal

In modeling information flow in a distributed system, the system itself is to be modeled as a information system $\mathcal{L}$. The constraints of the distributed logic $\text{Log}(\mathcal{L})$ model the available regularities of the system. The normal tokens of $\text{Log}(\mathcal{L})$ model the instances of the system to which the constraints are guaranteed to apply by virtue of the structure of the system.

It is difficult to say anything very informative about the theory of $\text{Log}(\mathcal{L})$ in general, because the various logics can interact in complicated ways. We can, however, describe the normal tokens of the logic.

Proposition 15.3. *A token $\{c_i\}_{i \in I}$ of $\text{Log}(\mathcal{L})$ is normal if and only if each c_i is a normal token of $\mathfrak{L}_i$ and $c_i = f(c_j)$ whenever there is an informorphism of the system of the form $f : A_i \rightleftarrows A_j$.*

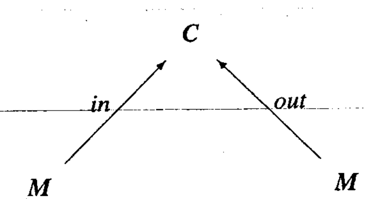
Proof. Using the notation of Definition 15.2, we first note that the normal tokens of $\mathfrak{L}$ consist of those tokens c of $\boldsymbol{C}$ such that $g_i(c) \in N_{\mathfrak{L}_i}$ for each i. From the definition of limit, the tokens of $\boldsymbol{C}$ are those $\{c_i\}_{i \in I}$ such that $c_i = f(c_j)$ for each infomorphism $f : \boldsymbol{A}_i \rightleftarrows \boldsymbol{A}_j$ of the system. But the normal tokens of Log($\mathcal{L}$) consist of the tokens $\{c_i\}_{i \in I} \in \text{tok}(\boldsymbol{A})$ of the form $g^{\check{}}(c)$ such that $c \in N_{\mathfrak{L}}$ and $g^{\check{}}$ is the identity on the tokens of $\boldsymbol{C}$. □

In applying this definition to a distributed system, we typically have a single logic on either the core of a channel or on one of its components. In such a case, we use the *a priori* logics on the other classifications of the system in order to be able to consider the distributed system as an information system. (One can think of the logic $\text{Log}_C(\boldsymbol{D})$, studied in Lecture 14, as a special case of the latter; use Log($\boldsymbol{P}$) together with AP($\boldsymbol{D}$) and AP($\boldsymbol{C}$) to get a logic on the sum, and then restrict the result to the classification $\boldsymbol{D}$.)

15.2 File Copying: An Example

If Judith copies a file, say, `adventure.tex`, from the Macintosh in her office to the one at home, using Apple Remote Access, she expects the copied file to have many of the same properties as the file copied. It should have the same contents, the same time stamp, the same icon, and so forth. Other properties will be different, like where it is in memory or what color of label it is given. Of course sometimes things go wrong and file copying does not work as expected. Our goal in this section is to show how we can look at this kind of information flow in terms of distributed logics. We will greatly simplify things, of course, as befits an illustrative example.

The distributed system $\mathcal{FC}_0$ ("$\mathcal{FC}$" for "file copying") that interests us is depicted as follows:



The tokens of $\boldsymbol{M}$ are instances of files on computers spread around the campus network. The tokens of $\boldsymbol{C}$ are events of successfully copying a file from one computer to another. The types of $\boldsymbol{M}$ consist of three types α, β, and γ, the first two of which are supposed to represent typical properties of files such as contents, time stamp, icon type, the kinds of things one wants preserved from a file to any copy of the file. We assume that α and β are independent properties of file tokens, things like contents and time stamp. The type γ is the type of being protected. Protected files should be prevented from being copied. The types of $\boldsymbol{C}$ include α^{in}, α^{out}, β^{in}, β^{out}, γ^{in}, γ^{out}, as well as some other types, typified by the type δ, that are not definable as properties of the input or output. These classifications are specified below, along with their infomorphisms. (We should really have a lot more tokens around to get the full flavor of the idea. Also, having more tokens around would help alleviate another problem that will come up. But having hundreds, thousands, or millions of tokens would make the classification tables hard to print, let alone read, so we ask the reader to imagine a lot more tokens in both classifications.)

$\models_M$	α	β	γ
m_1	0	1	0
m_2	1	1	1
m_3	1	0	1
m_4	1	1	1
m_5	1	0	1
m_6	0	1	1

$\models_C$	α^{in}	α^{out}	β^{in}	β^{out}	γ^{in}	γ^{out}	δ
c_1	1	1	1	1	1	1	1
c_2	1	1	0	0	1	1	0
c_3	1	0	1	1	1	1	1

transaction c	file copied ($=in^{\check{}}(c)$)	resulting file ($=out^{\check{}}(c)$)
c_1	m_2	m_4
c_2	m_3	m_5
c_3	m_4	m_6

The infomorphisms *in* and *out* are as suggested by the names we have given the types. It is clear from examination of these tables that these functions are indeed infomorphisms, and so the result is a channel. We think of it as a channel from the $\boldsymbol{M}$ on the left to that on the right, though we could equally well use it as a channel in the opposite direction. The table of connections (transactions) shows that the file token m_4 is a copy of m_2, m_5 is a copy of m_3, and m_6 is a copy of m_4. Notice also that the protected file token m_1 is not connected to anything, so is not copied nor is it a copy of any other file. One would have to unprotect it, thereby getting a different file token, before it could be copied. Notice also that the transaction c_3 was not totally successful; it produced the file m_6, but the property α of the copied file m_4 does not hold of m_6. We want to be able to account for facts like

$m_2 \models_M \beta$ carries the information that $m_4 \models_M \beta$.

According to our proposal, this should be possible to explain in terms of a canonical logic on $M + M$ such that $\langle m_2, m_4\rangle$ is a normal token of the logic, and $\beta^{in} \vdash \beta^{out}$ is a constraint of the logic.[1]

Given this channel, we can distribute any logic $\mathfrak{L}$ on C to the sum $M + M$. At this point, though, there are only two obvious logics available to us, the *a priori* logic AP(C) and the natural logic Log(C). Let us examine in turn what happens if we distribute these logics.

Example 15.4. Distributing the *a priori* logic AP(C) does not give the desired results. The resulting logic on $M + M$ has as normal tokens the pairs $\langle m_2, m_4\rangle$, $\langle m_3, m_5\rangle$, and $\langle m_4, m_6\rangle$. Because the infomorphisms *in* and *out* of the system have disjoint ranges in typ(C), the theory of this logic is just the smallest regular theory on the types of $M + M$. Hence $\beta^{in} \nvdash \beta^{out}$ so we do not have $m_2 \vDash_M \beta$ carrying the information about its copy m_4 that $m_4 \vDash_M \beta$, let alone any of the more subtle results we are after.

Example 15.5. Distributing the sound and complete Log(C) to $M + M$ is more successful. The normal tokens are the same as before, but the theory is much richer. Recalling that the rows of the classification table for A can be thought of as specifying its set of consistent partitions, we see that this logic contains the following constraints:

$$\beta^{in} \vdash_C \beta^{out}, \quad \beta^{out} \vdash_C \beta^{in}, \quad \vdash_C \gamma^{in}, \quad \vdash_C \gamma^{out}.$$

Consequently, relative to this logic we *do* have $m_2 \vDash_M \beta$ carrying the information that $m_4 \vDash_M \beta$ (and vice versa), and we get for free the information that both tokens are of type γ.

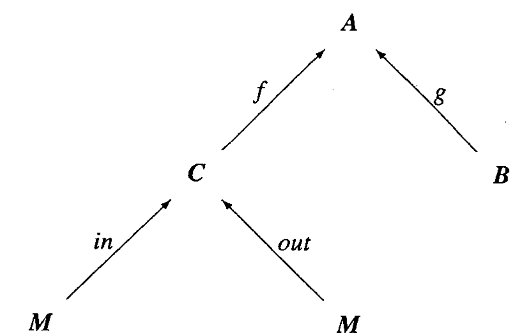
There are some disturbing features of the logic obtained by distributing Log(C), however, features that come from assuming complete information about the classification C. First, because of the faulty transaction c_3, $\alpha^{in} \nvdash_C \alpha^{out}$, so $m_4 \vDash_M \alpha$ does not carry the (mis)information that $m_6 \vDash_M \alpha$, and thus fails to reflect a mistake that users may justifiably make. Worse, we also do not find that $m_2 \vDash_M \alpha$ carries the information that $m_4 \vDash_M \alpha$. Using this logic, even one exceptional transaction out of millions can wreck information flow even for successful transactions. This is just the kind of result we want to avoid.

A different sort of problem comes from accidental generalizations captured by Log(C). For example, looking at the classification table, we see that as it happens $\beta^{in} \vdash_C \alpha^{in}$. This would imply, for example, that $m_2 \vDash_M \beta$ carries

[1] We are here using the same notation β^{in} for the copy of β in the sum and for the translation of β into the types of C. This should not cause any confusion.

the information that $m_2 \vDash_M \alpha$, which seems wrong. This information is carried by $m_4 \vDash_M \alpha$ but surely not by $m_2 \vDash_M \beta$; look at m_1. The problem, of course, is that although we said that α and β were independent properties of files, there are no tokens in the classification of transactions that bear witness to this independence.

To rectify these problems, we can modify our distributed system $\mathcal{FC}_0$, either by postulating a third logic on C or by adding classifications and infomorphisms to the system. We pursue the latter approach first. Expand $\mathcal{FC}_0$ to a larger system $\mathcal{FC}$ by adding two classifications A and B as depicted below.



The classification A is the same as C except it throws out the faulty connection c_3:

$\vDash_A$	α^{in}	α^{out}	β^{in}	β^{out}	γ^{in}	γ^{out}	δ
c_1	1	1	1	1	1	1	0
c_2	1	1	0	0	1	1	1

Here f is the identity on types and on the tokens of A. Before describing B, let us see what happens if we distribute Log(A).

Example 15.6. If we distribute Log(A), the normal tokens are what we want: $\langle m_2, m_4\rangle$ and $\langle m_3, m_5\rangle$. Also, looking at the classification table, we see that

$$\alpha^{in} \vdash_A \alpha^{out}$$

so that relative to this logic, $m_2 \vDash_M \alpha$ carries the information that $m_4 \vDash_M \alpha$, as desired. Moreover, having this constraint does not mean that $m_4 \vDash_M \alpha$ carries the information that $m_6 \vDash_M \alpha$ because $\langle m_4, m_6\rangle$ is not a normal token. However, distributing Log(A) still makes the false claim concerning information flow that $m_2 \vDash_M \beta$ carries the information that $m_2 \vDash_M \alpha$.

To take care of the overgeneration due to accidental generalizations, we add idealized tokens in the classification B defined below:

$\models_B$	α^{in}	α^{out}	β^{in}	β^{out}	γ^{in}	γ^{out}
c_1	1	1	1	1	1	1
c_2	1	1	0	0	1	1
n_1	0	0	1	1	1	1
n_2	0	0	0	0	1	1

The tokens n_1 and n_2 are idealized tokens used to represent unactualized possibilities embodied in our claim of the independence of α and β. The infomorphism g is the identity on types of B and on tokens of A.

Example 15.7. Consider the logic on $M + M$ obtained by distributing Log(B). We compute this logic in two steps. First, let $\mathfrak{L} = f^{-1}[g[\mathrm{Log}(B)]]$. This is a local logic on C with normal tokens c_1 and c_2. Because both f and g are type-identical, the theory of $\mathfrak{L}$ is Th(B). We claim that Th(B) is the regular closure of the following six constraints:

$$\alpha^{in} \vdash_C \alpha^{out}, \quad \alpha^{out} \vdash_C \alpha^{in}, \quad \vdash_C \gamma^{in},$$
$$\beta^{in} \vdash_C \beta^{out}, \quad \beta^{out} \vdash_C \beta^{in}, \quad \vdash_C \gamma^{out}.$$

To see this, we apply Proposition 9.18. First note that each of these constraints is valid in B. Next, note that any row of 0s and 1s that disagrees with those present in the classification table for B invalidates at least one of these constraints. For example, the row

$\models_B$	α^{in}	α^{out}	β^{in}	β^{out}	γ^{in}	γ^{out}
c_1	1	1	1	0	1	1

violates the constraint $\beta^{in} \vdash_B \beta^{out}$. If we now distribute $\mathfrak{L}$ over $M + M$, we get just the intuitively valid constraints; the normal tokens consist of the pairs $\langle m_2, m_4\rangle$ and $\langle m_3, m_5\rangle$.

We see that there are two ways to get at what seems like the natural information-theoretic logic on $M + M$. One is by distributing the natural logic Log(B) over the system $\mathcal{FC}$. The other is by distributing the logic $\mathfrak{L}$ $(= f^{-1}[g[\mathrm{Log}(B)]])$ over $\mathcal{FC}_0$. This corresponds to a information system with $\mathfrak{L}$ on C and AP(M) on each copy of M. These approaches give us the same logic on $M + M$, but they give us somewhat different ways of looking at information flow.

15.3 The Distributed Logic of a Distributed System

We now turn from our little example to some special cases of the general construction. We begin with the pure case, where we have a distributed system $\mathcal{A}$ with no logics on any of the classifications. Put in terms of information systems, we are assuming that we have an information system where each classification has its *a priori* logic.

Proposition 15.8. *Let $\mathcal{A}$ be a distributed system with classifications* cla($\mathcal{A}$) $=$ $\{A_i\}_{i\in I}$ *and infomorphisms* inf($\mathcal{A}$). *The systemwide logic* Log($\mathcal{A}$) *can be characterized as follows.*

1. *The classification of the logic is $\sum_{i\in I} A_i$.*
2. *The theory of the logic is the regular closure of the set of constraints of the forms*

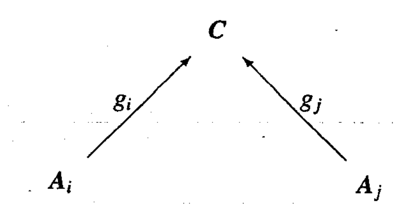
$$\alpha \vdash f(\alpha) \quad \textit{and} \quad f(\alpha) \vdash \alpha$$

for each $f : A_i \rightleftarrows A_j$ of the system and each $\alpha \in$ typ(A_i).

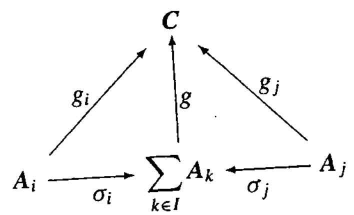
3. *The normal tokens are just those indexed families $c = \{c_i\}_{i\in I}$ of tokens that correspond to a global choice of a token from each component so that the pieces respect the whole-part relationships of the system. That is, $c \in N_{\mathfrak{L}}$ if and only if whenever $f : A_i \rightleftarrows A_j$ and $c_j \in$ tok(A_j), then $c_i = f(c_j)$.*

Proof. In stating this result we have tacitly assumed that the types of the various component classifications are disjoint. If not, they need to be replaced by their disjoint copies. We continue to make this notationally simplifying assumption in the remainder of this lecture. The proof is just a matter of unwinding our earlier definition where all the component logics are *a priori* logics. □

Let us depict the limit lim $\mathcal{A}$ of our distributed system $\mathcal{A}$ as follows:



Using the fundamental property of sums, we obtain the following commuting diagram, where we write σ_i for σ_{A_i} and g for $\sum_{k\in I} g_k$:



Using this notation, we can now explicate the relationship between the limit and the systemwide logic.

Theorem 15.9. *Let $\mathcal{A}$ be a distributed system as above and let $\mathfrak{L} = \mathrm{AP}(C)$ on the core of* $\lim \mathcal{A}$.

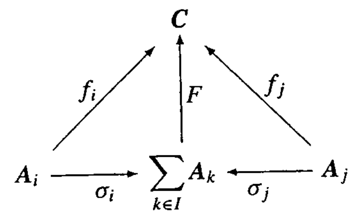
1. $\mathfrak{L} = \mathrm{Lind}(\mathrm{Log}(\mathcal{A}))$.
2. *If* $\tau : \mathrm{Log}(\mathcal{A}) \rightleftarrows \mathrm{Lind}(\mathrm{Log}(\mathcal{A}))$ *is the quotient logic infomorphism, then for each* i, $g_i = \tau\sigma_i$.
3. $\mathrm{Log}(\mathcal{A}) = g^{-1}[\mathfrak{L}]$.

Proof. The proofs of (1) and (2) are clear by examining the definition of the limit in Lecture 6 and of the Lindenbaum logic in Lecture 12. To prove (3), we first recall that $g^{-1}[\mathfrak{L}]$ is the largest logic $\mathfrak{L}'$ on A such that $g : \mathfrak{L}' \rightleftarrows \mathfrak{L}$ is a logic infomorphism. However, it is clear that $g : \mathrm{Log}(\mathcal{A}) \rightleftarrows \mathfrak{L}$ is a logic infomorphism, so $\mathrm{Log}(\mathcal{A}) \sqsubseteq g^{-1}[\mathfrak{L}]$. Thus we need only show that $g^{-1}[\mathfrak{L}] \sqsubseteq \mathrm{Log}(\mathcal{A})$. Because $\mathfrak{L}$ is sound, the condition on tokens is trivial. So we need only show that if $\Gamma \vdash_{g^{-1}[\mathfrak{L}]} \Delta$, then $\Gamma \vdash_{\mathrm{Log}(\mathcal{A})} \Delta$, for all $\Gamma, \Delta \subseteq \mathrm{typ}(A)$. Assume $\Gamma \vdash_{g^{-1}[\mathfrak{L}]} \Delta$. Then $g[\Gamma] \vdash_{\mathfrak{L}} g[\Delta]$. Thus there are types $\alpha \in \Gamma, \beta \in \Delta$ such that $F(\alpha) = F(\beta)$, because $\mathfrak{L}$ is the *a priori* logic on C. But this means that $[\alpha]_R = [\beta]_R$, where R is the relation used in taking the quotient in the construction of C. But then an inductive proof (on the length of a chain of types from α to β using R and R^{-1}) shows that $\alpha \vdash_{\mathrm{Log}(\mathcal{A})} \beta$, as desired. □

15.4 Distributing a Logic Along a Channel

Suppose we are given an information channel $\mathcal{C} = \{f_i : A_i \rightleftarrows C\}_{i\in I}$ and a logic $\mathfrak{L}$ on the core classification C. We want to define a local logic on the sum $\sum_{i\in I} A_i$ that represents the reasoning about relations among the components justified by the logic $\mathfrak{L}$ on the core. We call this the distributed logic of $\mathcal{C}$ generated by $\mathfrak{L}$ and denote it by $\mathrm{DLog}_{\mathcal{C}}(\mathfrak{L})$.

To define this logic, recall the diagram of the channel together with the canonical embedding of the sum into the core of the channel.



We can characterize this logic in terms of our previous construction by considering the whole channel as a distributed system. However, this gives us a logic on $\sum_{k\in I} A_k + C$, whereas what we want is a logic on $\sum_{k\in I} A_k$. Of course we can easily extract a logic on the first summand from that on the sum. We can characterize the logic directly as $F^{-1}[\mathfrak{L}]$, where F is as above.

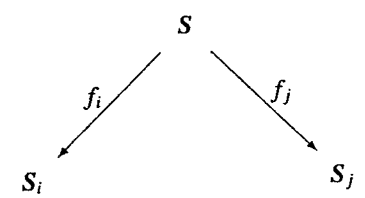
If the logic $\mathfrak{L}$ on the core of a channel $\mathcal{C}$ is complete, then the distributed logic $\mathrm{DLog}_{\mathcal{C}}(\mathfrak{L})$ is complete, simply because taking inverse images preserves completeness. Note, though, that $\mathrm{DLog}_{\mathcal{C}}(\mathfrak{L})$ is not in general a sound logic, even if $\mathfrak{L}$ is sound. It is only guaranteed to be sound on the range $F^{\vee}$, that is, on those tokens of the sum that are sequences of projections of a normal token of $\mathfrak{L}$. In other words, the normal tokens of the distributed logic $\mathrm{DLog}_{\mathcal{C}}(\mathfrak{L})$ consist of those sequences of components that are connected together by some normal token of the whole system.

15.5 The Distributed Logic of a State-Space System

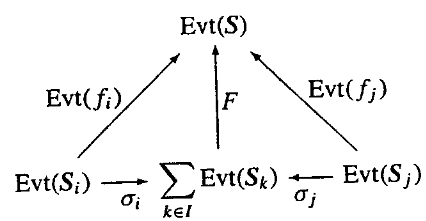
In Lecture 3, we investigated an example of an information channel whose core was the event classification of a state space S. There we were using the state-space logic $\mathrm{Log}(S)$. This is frequently a very convenient tool. In this section, we explore what it means to distribute this logic over a state-space system.

Suppose we are given a state-space system $\mathcal{S} = \{f_i : S \rightrightarrows S_i\}_{i\in I}$. We want to define a local logic on the sum $\sum_{i\in I} \mathrm{Evt}(S_i)$ that captures the way we can infer partial information about the state of some components given partial information about the state of some other components and the fact that they are connected by some token of the system. We call this the distributed logic generated by $\mathcal{S}$, and denote it by $\mathrm{DLog}_{C}(\mathcal{S})$.

We already have one logic on the sum $\sum_{i \in I} \mathrm{Evt}(S_i)$ of course, namely, the sum of the logics generated by the component state spaces, that is, $\sum_{i \in I} \mathrm{Log}(S_i)$. This logic ignores the connections furnished by the core state space S, however. To take advantage of this state space, we simply distribute the logic Log(S) to $\sum_{i \in I} \mathrm{Evt}(S_i)$ by means of a natural infomorphism. Recall from Lecture 8 the following construction. Start with the system $\mathcal{S}$ depicted by the following diagram:



Applying the operator Evt to this diagram and using the fundamental property of sums, we obtained the following commuting diagram, where for the sake of readability, we write σ_i for $\sigma_{\mathrm{Evt}(S_i)}$ and F for $\sum_{k \in I} \mathrm{Evt}(f_k)$:



The infomorphism F allows us to pull back any logic $\mathfrak{L}$ on Evt(S) to a logic $F^{-1}[\mathfrak{L}]$ on the classification $\sum_{k \in I} \mathrm{Evt}(S_k)$.

Definition 15.10. Given a state-space system $\mathcal{S} = \{f_i : S \rightrightarrows S_i\}_{i \in I}$, the *distributed logic* DLog($\mathcal{S}$) of $\mathcal{S}$ is $F^{-1}[\mathrm{Log}(S)]$, where F is as above.

Notice that if the core state space S of $\mathcal{S}$ is complete, then this logic is complete, simply because taking inverse images preserves completeness. When using any state space the presumption is that it is complete; hence the presumption carries over to the distributed logic DLog($\mathcal{S}$).

Note also, though, that DLog($\mathcal{S}$) is not in general a sound logic. It is only guaranteed to be sound on the range of $F^{\vee}$, that is, on those tokens of DLog($\mathcal{S}$) that are sequences of projections of a token of S. In other words, the normal tokens of the distributed logic DLog($\mathcal{S}$) consist of those sequences of components that are connected together by some token of the whole system. These tokens satisfy all the constraints of the logic for principled reasons: they have to by virtue of the state-space logic on their connections. There might be other tokens around that satisfy the constraint of the logic, but if so, they do so by accident.

There is a simple, explicit characterization of the consequence relation of DLog($\mathcal{S}$). To make it easier to read, we adopt the following notation. Given any set Γ of types in our logic, write $\Gamma_k = \{X \in \Gamma \mid X \in \mathrm{typ}(\mathrm{Evt}(S_k))\}$, a set of types in Evt($S_k$). Thus $\Gamma = \bigcup_{k \in I} \Gamma_k$.

Theorem 15.11. *Let $\mathcal{S} = \{f_i : S \rightrightarrows S_i\}_{i \in I}$ be a state-space system. For any sets Γ, Δ of types of the distributed logic* DLog($\mathcal{S}$), *the following are equivalent:*

1. $\Gamma \vdash_{\mathrm{DLog}(\mathcal{S})} \Delta$.
2. *For each state σ of the core of $\mathcal{S}$ and each $k \in I$, $\Gamma_k, \{f_k(\sigma)\} \vdash_{\mathrm{Log}(S_k)} \Delta_k$.*
3. *For each state σ of the core of $\mathcal{S}$ and each $k \in I$, if $f_k(\sigma) \in \bigcap \Gamma_k$, then $f_k(\sigma) \in \bigcup \Delta_k$.*

The normal tokens of DLog($\mathcal{S}$) *consist of those sequences of components that are connected by some token c of the core S of $\mathcal{S}$. The logic is complete if the state space S is complete.*

Proof. To prove the first claim, note that the following are equivalent:

$$\Gamma \vdash_{\mathrm{DLog}(\mathcal{S})} \Delta.$$
$$F[\Gamma] \vdash_{\mathrm{Log}(S)} F[\Delta].$$
$$\{F(X) \mid X \in \Gamma\} \vdash_{\mathrm{Log}(S)} \{F(Y) \mid Y \in \Delta\}.$$
$$\bigcup_{k \in I}\{F(X) \mid X \in \Gamma_k\} \vdash_{\mathrm{Log}(S)} \bigcup_{k \in I}\{F(Y) \mid Y \in \Delta_k\}.$$
$$\bigcup_{k \in I}\{f_k^{-1}[X] \mid X \in \Gamma_k\} \vdash_{\mathrm{Log}(S)} \bigcup_{k \in I}\{f_k^{-1}[Y] \mid Y \in \Delta_k\}.$$

For each state σ of the core of $\mathcal{S}$ and each $k \in I$, if $f_k(\sigma) \in \bigcap \Gamma_k$, then $f_k(\sigma) \in \bigcup \Delta_k$

For each state σ of the core of $\mathcal{S}$ and each $k \in I$, $\Gamma_k, \{f_k(\sigma)\} \vdash_{\mathrm{Log}(S_k)} \Delta_k$.

Each item in our theorem is in this list. The final two sentences merely summarize the earlier discussion. □

Exercises

15.1. The fundamental dogma of molecular biology asserts that information flows from DNA to RNA to protein but not in the reverse direction. Model this information flow as an information system and explain a sense in which information flow is one directional.

15.2. Using the results on moving logics, generalize the results on limits of distributed systems to information systems. The first step is to generalize our previous notions from classifications (essentially natural logics) to arbitrary local logics.

Lecture 16

Logics and State Spaces

In Lecture 8 we saw how to construct state spaces from classifications and vice versa. In Lecture 12 we saw how to associate a canonical logic Log(S) with any state space S. In this lecture we study the relation between logics and state spaces in more detail. Our aim is to try to understand how the phenomena of incompleteness and unsoundness get reflected in the state-space framework. We will put our analysis to work by exploring the problem of nonmonotonicity in Lecture 19.

16.1 Subspaces of State Spaces

Our first goal is to show that there is a natural correspondence between the subspaces of a state space S and logics on the event classification of S. We develop this correspondence in the next few results.

Definition 16.1. Let S be a state space. An S-logic is a logic $\mathfrak{L}$ on the event classification Evt(S) such that Log(S) $\sqsubseteq \mathfrak{L}$.

The basic intuition here is that an S-logic should build in at least the theory implicit in the state-space structure of S. We call a state σ of S *$\mathfrak{L}$-consistent* if $\{\sigma\} \nvdash_{\mathfrak{L}}$ and let $\Omega_{\mathfrak{L}}$ be the set of $\mathfrak{L}$-consistent states.

Proposition 16.2. *If S is a state space and $\mathfrak{L}$ is an S-logic, then* $\vdash_{\mathfrak{L}} \Omega_{\mathfrak{L}}$. *Indeed,* $\Omega_{\mathfrak{L}}$ *is the smallest set of states such that* $\vdash_{\mathfrak{L}} \Omega_{\mathfrak{L}}$.

Proof. To prove the first claim, let $\langle \Gamma, \Delta \rangle$ be any partition of the types of Evt(S) with $\Omega_{\mathfrak{L}} \in \Delta$. We need to see that $\Gamma \vdash_{\mathfrak{L}} \Delta$. Because Log($S$) $\sqsubseteq \mathfrak{L}$, we need only check that every state σ that is in every $X \in \Gamma$ is in some $X \in \Delta$. If not, then $\sigma \notin \Omega_{\mathfrak{L}}$, because $\Omega_{\mathfrak{L}} \in \Delta$. Hence σ is inconsistent in $\mathfrak{L}$. But $\{\sigma\} \in \Gamma$ because it cannot be in Δ, so $\Gamma \vdash_{\mathfrak{L}} \Delta$ by Weakening. To prove the second claim, suppose that $\vdash_{\mathfrak{L}} \Omega'$. We want to prove that $\Omega_{\mathfrak{L}} \subseteq \Omega'$. Supposing this is not the case, let $\sigma \in \Omega_{\mathfrak{L}} - \Omega'$. If we can prove $\{\sigma\} \vdash_{\mathfrak{L}}$, we will have our desired contradiction. We want to use Partition, so let $\langle \Gamma, \Delta \rangle$ be a partition with $\{\sigma\} \in \Gamma$. We want to show that $\Gamma \vdash_{\mathfrak{L}} \Delta$. Because $\vdash_{\mathfrak{L}} \Omega'$, it suffices, by Cut, to prove that $\Gamma, \Omega' \vdash_{\mathfrak{L}} \Delta$. Because $\mathfrak{L}$ is an S-logic, it suffices to show $\Gamma, \Omega' \vdash_{\mathrm{Log}(S)} \Delta$. But this is obvious, because Γ contains an element disjoint from Ω', namely $\{\sigma\}$. □

Definition 16.3. Let S be a state space and let $\mathfrak{L}$ be an S-logic. The *subspace* $S_{\mathfrak{L}}$ *of* S *determined by* $\mathfrak{L}$ has $\Omega_{\mathfrak{L}}$ for its set of states and has as tokens the set $N_{\mathfrak{L}}$ of normal tokens of $\mathfrak{L}$.

Justification. We need to check that the state of each normal token of $\mathfrak{L}$ is a member of $\Omega_{\mathfrak{L}}$. This follows from $\vdash_{\mathfrak{L}} \Omega_{\mathfrak{L}}$. □

Proposition 16.4. *Let S be a state space and let $\mathfrak{L}$ be an S-logic. Then $\Gamma \vdash_{\mathfrak{L}} \Delta$ if and only if $\Gamma, \Omega_{\mathfrak{L}} \vdash_{\mathrm{Log}(S)} \Delta$.*

Proof. The direction from right to left follows from Proposition 16.2 and Finite Cut. To prove the converse, suppose that $\Gamma \vdash_{\mathfrak{L}} \Delta$. We need to prove that every state $\sigma \in \Omega_{\mathfrak{L}}$, if $\sigma \in X$ for every $X \in \Gamma$, then $\sigma \in Y$ for some $Y \in \Delta$. Suppose σ is a counterexample. We want to show that σ is inconsistent in $\mathfrak{L}$, contradicting the fact that it is in $\Omega_{\mathfrak{L}}$. To prove this, let $\langle \Gamma', \Delta' \rangle$ be a partition with $\{\sigma\} \in \Gamma'$ such that $\Gamma' \nvdash_{\mathfrak{L}} \Delta'$. From this it is easy to see that

$$\Gamma' = \{X \mid \sigma \in X\},$$
$$\Delta' = \{X \mid \sigma \notin X\}.$$

But then $\Gamma \subseteq \Gamma'$ and $\Delta \subseteq \Delta'$, so $\Gamma' \vdash_{\mathfrak{L}} \Delta'$ by Weakening. □

Proposition 16.5. *If S is a state space and $\mathfrak{L}_1, \mathfrak{L}_2$ are S-logics, then $\mathfrak{L}_1 \sqsubseteq \mathfrak{L}_2$ if and only if $S_{\mathfrak{L}_2} \subseteq S_{\mathfrak{L}_1}$. Hence $\mathfrak{L}_1 = \mathfrak{L}_2$ if and only if $S_{\mathfrak{L}_2} = S_{\mathfrak{L}_1}$.*

Proof. The second claim clearly follows from the first. To prove the first, assume $\mathfrak{L}_1 \sqsubseteq \mathfrak{L}_2$. Because $S_{\mathfrak{L}_1}$ and $S_{\mathfrak{L}_2}$ are both subspaces of S, all we need to check is that the tokens of the latter are a subset of the tokens of the former and that the states of the latter are a subset of the states of the former. The first is immediate from the assumption and the definitions of these spaces. As for states, the claim

is almost as obvious. If $\sigma \in \Omega_{\mathfrak{L}_2}$, then $\sigma \nvdash_{\mathfrak{L}_2}$. But by our assumption, $\sigma \nvdash_{\mathfrak{L}_1}$, and so $\sigma \in \Omega_{\mathfrak{L}_1}$.

Now assume $S_{\mathfrak{L}_2} \subseteq S_{\mathfrak{L}_1}$. Again, the inclusion of normal tokens is trivial, so we need only verify the inclusion of constraints. Suppose $\Gamma \vdash_{\mathfrak{L}_1} \Delta$. By the previous result, $\Gamma, \Omega_{\mathfrak{L}_1} \vdash_{\mathrm{Log}(S)} \Delta$. Hence $\bigcap \Gamma \cap \Omega_{\mathfrak{L}_1} \subseteq \bigcup \Delta$. But $\Omega_{\mathfrak{L}_2} \subseteq \Omega_{\mathfrak{L}_1}$ by assumption, so $\bigcap \Gamma \cap \Omega_{\mathfrak{L}_2} \subseteq \bigcup \Delta$. Hence $\Gamma, \Omega_{\mathfrak{L}_2} \vdash_{\mathrm{Log}(S)} \Delta$, and so $\Gamma \vdash_{\mathfrak{L}_2} \Delta$. □

Theorem 16.6. *Let S be any state space. The mapping*

$$\mathfrak{L} \mapsto S_{\mathfrak{L}}$$

is an order inverting bijection between the family of all S-logics and the family of all subspaces of S.

Proof. Given Proposition 16.5, all we need establish is that every subspace S_0 of S is of the form $S_{\mathfrak{L}}$ for some logic extending Log(S). Let the normal tokens of $\mathfrak{L}$ be the tokens of S_0. Define a consequence relation on Evt(S) by $\Gamma \vdash_{\mathfrak{L}} \Delta$ if and only if $\Gamma, \mathrm{typ}(S_0) \vdash_{\mathrm{Log}(S)} \Delta$. This is easily seen to be regular. All we need do is to check that for any state σ of S, $\sigma \vdash_{\mathfrak{L}}$ if and only if $\sigma \notin \mathrm{typ}(S_0)$. This is routine. □

These results show that the state-space analog of a logic on a classification is that of a subspace of the given state space. The tokens of the subspace correspond to normal tokens of the logic, the states of the subspace correspond to the constraints of the logic. Here is another way in which this analogy holds.

Definition 16.7. Let S be a state space, S_0 a subspace of S. S_0 is *sound* in S if $\mathrm{tok}(S_0) = \mathrm{tok}(S)$.

Proposition 16.8. *Let S be a state space. An S-logic $\mathfrak{L}$ is sound if and only if the associated state space $S_{\mathfrak{L}}$ is sound.*

Proof. We leave this as an exercise. □

The following shows that the concepts of completeness and soundness for state spaces behave as one would expect, given the above correspondence.

Proposition 16.9. *Let $f : S \rightrightarrows S'$ be a state-space projection.*

1. *If S_0 is a complete subspace of S, then $f[S_0]$ is a complete subspace of S'.*

2. *If S_1 is a sound subspace of S', then $f^{-1}[S_1]$ is a sound subspace of S.*

Proof. The proof is a routine verification. □

16.2 From Local Logics to State Spaces

We now complete the correspondence between local logics and state spaces by showing how any local logic $\mathfrak{L}$ gives rise to a canonical state space Ssp($\mathfrak{L}$). Recall (from Lecture 8) the state space Ssp(A) generated by a classification A. This space has the same tokens as A; states are arbitrary partitions of the types of A. The state of a token is its state description.

Definition 16.10. Let $\mathfrak{L}$ be a local logic on a classification A.

1. The state space Ssp($\mathfrak{L}$) *generated by* $\mathfrak{L}$ is the subspace of Ssp(A) whose tokens are the normal tokens of $\mathfrak{L}$ and whose types are the $\mathfrak{L}$-consistent partitions.
2. Given a logic infomorphism $f : \mathfrak{L}_1 \rightleftarrows \mathfrak{L}_2$, let Ssp($f$) be the state-space projection from Ssp($\mathfrak{L}_2$) to Ssp($\mathfrak{L}_1$) that is the restriction of Cla(f) to Ssp($\mathfrak{L}_2$).

Justification. Because every normal token has a consistent state description, this does indeed define a subspace of Ssp(A). It is easy to check that things work properly on maps. □

Proposition 16.11. *Let $\mathfrak{L}$ be a local logic on a classification A.*

1. *$\mathfrak{L}$ is sound if and only if* Ssp($\mathfrak{L}$) *is a sound subspace of* Ssp(A).
2. *$\mathfrak{L}$ is complete if and only if* Ssp($\mathfrak{L}$) *is a complete state space.*

We have the following analog of Theorem 16.6.

Theorem 16.12. *Let A be any classification and let S = Ssp(A) be its associated state space. The mapping*

$$\mathfrak{L} \mapsto \text{Ssp}(\mathfrak{L})$$

is an order inverting bijection between the set of logics on A and the set of subspaces of S.

Proof. The proof is very similar to the proof of Theorem 16.6 and is entirely straightforward. □

Exercises

16.1. By Theorem 16.6, different subspaces correspond to different logics. This raises a question as to the relationship between images and inverse images of subspaces, on the one hand, and logics, on the other. Let $f : S \rightrightarrows S'$ be a projection, and let S_1 and S_2 be subspaces of S and S', respectively. Recall that Evt(f) : Evt(S') $\rightleftarrows$ Evt(S) is an infomorphism. Hence the image under Evt(f) of a logic on Evt(S') is a logic on Evt(S) and the inverse image of a logic on Evt(S) is a logic on Evt(S'). Prove the following identities, where we write Log(S_1) for the S-logic $\mathfrak{L}$ that corresponds to S_1 under the bijection of Theorem 16.6:

1.
$$\text{Log}(f[S_1]) = \text{Evt}(f)^{-1}[\text{Log}(S_1)].$$
2.
$$\text{Log}(f^{-1}[S_2]) = \text{Evt}(f)[\text{Log}(S_2)].$$

16.2. Prove Proposition 16.11.

16.3. Let $f : A \rightleftarrows B$ be an infomorphism. By Theorem 16.12, subspaces of these state spaces correspond to logics on A and B, respectively. This raises a question as to the relationship between images and inverse images of subspaces, on the one hand, and logics, on the other. Recall that Ssp(f) : Ssp(B) $\rightrightarrows$ Ssp(A) is a state-space projection. Hence the image under Ssp(f) of a subspace of Ssp(B) is a subspace of Ssp(A) and the inverse image of a subspace of Ssp(A) is a subspace of Ssp(B). Let $\mathfrak{L}_1$ and $\mathfrak{L}_2$ be logics on A and B, respectively. Prove the following identities:

1.
$$\text{Ssp}(f[\mathfrak{L}_1]) = \text{Ssp}(f)^{-1}[\text{Ssp}(\mathfrak{L}_1)].$$
2.
$$\text{Ssp}(f^{-1}[\mathfrak{L}_2]) = \text{Ssp}(f)^{-1}[\text{Ssp}(\mathfrak{L}_2)].$$

Part III

Explorations

Lecture 17

Speech Acts

In this lecture we want to give a simple application of classifications and infomorphisms to analyze J. L. Austin's four-way distinction in "How to talk: Some simple ways" (Austin, 1961). The material in this lecture follows Lecture 4 and is not needed elsewhere in the book.

17.1 Truth-Conditional Semantics and Speech Acts

The theory of speech acts owes its origins to Austin's (1961) work. This theory is challenging in a couple of ways. First, Austin's paper is one of his more difficult. It is just hard to figure out what he is saying. Second, the theory of speech acts poses a challenge to certain kinds of semantic theories, and the types of speech acts discussed by Austin in his paper illustrate the challenge very clearly. Austin is saying that there are at least four distinct things a person can be doing with a true utterance as simple as "Figure 4 is a triangle." One might say that he is arguing that such an utterance can have at least four distinct types of content. The difference is not reflected in the truth conditions of the utterance, but in something else entirely. If this is right, it seems to pose a special problem for semantic theories that try to explicate sentence meaning in terms of truth conditions.

Austin's paper has a special relevance to our project, as well. This work grew out of attempting to flesh out ideas about constraints presented in *Situations and Attitudes* (Barwise and Perry, 1983), a book that owes much to Austin's general approach to language. It thus seems appropriate to try to bring our insights back to bear on Austin's theory.

Our hope, then, is to do two things in this lecture. One is to try to repay a debt to Austin by using the theory of classifications and infomorphisms to shed some light on his difficult paper. More importantly, though, we hope to suggest initial directions for how channel theory might be used to contribute to a logical approach to speech acts. We emphasize that this lecture, like all of Part III, is intended to be thought provoking and highly tentative and not anything like a full theory of speech acts.

17.2 Austin's Model

In order to elucidate the distinctions he wants to make, Austin begins with a very simplified model of language in use, what he calls the model S_0. Having made the distinctions, he goes on to complicate the model in various ways. We will stick to Austin's S_0 model. Austin assumes the following setup.

The Described Situation

On the world side of the his language/world model, Austin assumes there is a set of *items* and that each item is of a unique *type*. Austin uses two running examples to illustrate his observations, one where items are particular color patches and types are colors, the other where items are drawn figures on a page and the types are geometric shapes like circle, triangle, square, rhombus, and so on. We will restrict our attention to the latter example. Thus, from our perspective, Austin assumes that the world consists of a classification ***Fig***, and his items are our tokens.

Language

Austin uses a very simple language to illustrate the ways we can talk about the described situation. In particular, in the S_0 model, Austin allows only sentences S of the form

$$t \text{ is a } N,$$

where t is a term that refers to some item and N is a word whose sense is some type. That is, Austin essentially assumes we are given a set *Names*, a set *Nouns*, a reference function $ref: Names \rightarrow \text{tok}(\boldsymbol{Fig})$, and a sense function $sense: Nouns \rightarrow \text{typ}(\boldsymbol{Fig})$. An object may have more than one name, and a type may be the sense of more than one noun, but in this model, at least, each name refers to a unique item in ***Fig*** and each noun has a unique sense, a type of ***Fig***.

To make things a little more definite, we assume that every name has the form Figure n for some numeral n. We assume that the set of nouns contains the expressions triangle, square, rhombus, pentagon, hexagon, septagon, and circle and that the function *sense* assigns these nouns their usual geometric senses. We assume that the set of sentences consists of the expressions

Figure n is a N,

where n is a numeral and N is any of the above nouns.

17.3 Austin's Four Speech Acts

In Austin's theory, utterances are more fundamental than sentences. In *Situations and Attitudes* (Barwise and Perry, 1983) this distinction was exploited to handle matters like tense, indexicals, who is being referred to by a given name, what sense of a noun is being used, which described situation is intended, and the like. In Austin's S_0, though, all such matters have been avoided, so one might think that we could deal simply with sentences. But this is not possible. Austin's model shows that even without the intrusion of these complexities, it is still possible to do more than one thing with a given sentence, necessitating the classification of utterances into different types of speech acts.

We thus suppose that there is another classification $\boldsymbol{U}$ of utterances. Each token is, intuitively, an utterance. The types of $\boldsymbol{U}$ are of two kinds, sentence types and speech-act types. The sentence types consist of all sentences of the displayed form, where $t \in Names$ and $N \in Nouns$. It is assumed that each token u is of a unique type. The speech-act types consist of four types: *placing*, *stating*, *instancing*, and *casting*. (That is all we need here, though for more sophisticated analyses one would add types to pick out the speaker, addressee, and so on.) Our aim here is to associate with each of these four types something it would be reasonable to call the "content" of a given utterance of that type.

Statings and Castings

We begin with the simplest of Austin's distinctions, statings versus castings.

Statings

If the sentence type of u is

$$t \text{ is a } N,$$

we write t_u for the name and N_u for the noun used in the utterance. At first sight, it would seem that an utterance u of "t is a N" could make only a single claim: that the item $ref(t_u)$ referred to is of the type $sense(N_u)$ in the world

Fig, that is, that

$$ref(t_u) \models_{Fig} sense(N_u).$$

Austin calls utterances that make this sort of claim *statings*, reasonably enough. Such a stating u is *true* if and only if $ref(t_u) \models_{Fig} sense(N_u)$. What other sorts of utterances could there be?

Austin discusses utterances in which names and nouns are misused, names to refer to items that they do not name, or nouns to connote a sense they do not have. But he discusses this only to set it aside as not being relevant to the four-way distinction he is after.

Castings

There are two ways of thinking of classifications. One may think of tokens as somehow more firmly anchored in the physical world and types as more abstract. But the tokens of a classification can be thought of as the givens, the things that need classifying. Types are the things we use to classify the tokens. Often these two notions march in step because it is more typical for the given to be given physically, but sometimes they do not. For example, if a classification $\mathbf{A}$ has physical tokens and abstract types, then the dual classification $\mathbf{A}^{\perp}$ (where types and tokens are interchanged) has physical types and abstract tokens.

Austin's notion of a casting is one where the tokens are more abstract. The basic idea of a casting is that one is given a type and is trying to "cast" it, as in a play; that is, one is looking for a token of that type. As Austin puts it, "To *cast* we have to find a sample to match this pattern to." If we think of the tokens of a classification as the givens, then this amounts to using the dual classification $\mathbf{Fig}^{\perp}$. That is, an utterance u is a *casting* if it asserts that

$$sense(N_u) \models_{Fig^{\perp}} ref(t_u).$$

Note that

$$ref(t_u) \models_{Fig} sense(N_u) \text{ if and only if } sense(N_u) \models_{Fig^{\perp}} ref(t_u),$$

so there is no difference of truth between a stating and a casting that use the same names and nouns. There is a difference in what the speaker is doing.

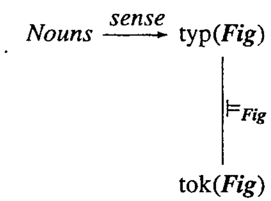
Placings and Instancings

Placings and instancings are like statings and castings, respectively, except in one key regard. In the latter pair, Austin says, the sense of the noun is taken for granted, whereas in the former pair it is not. The difference is most easily seen with examples of error. We look first at placings.

Placings

Suppose Tom says "Figure 3 is a square." One sort of error (misstating) is that Tom has gotten the shape of Figure 3 wrong, perhaps by misperceiving it. If Tom misstates that Figure 3 is a square, then he has mistaken the shape of Figure 3. By contrast, he misplaces Figure 3 in saying "Figure 3 is a square" if he has mistaken what one means by "square."

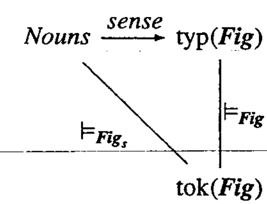
An easy way to capture this distinction is to use an induced classification. Consider the following diagram:



From these ingredients we obtain an induced classification $\mathbf{Fig}_s$ (the "s" to remind us that it is induced by $\mathbf{Fig}$ and the function *sense*) of the items by nouns as follows:

$$a \models_{Fig_s} N \quad \text{iff} \quad a \models_{Fig} sense(N)$$

This says that a figure is called an N if it has the shape $sense(N)$. This can be pictured as follows:



Another way of putting this is to require that *sense* (paired with the identity function on items) be an infomorphism $sense\colon \mathbf{Fig}_s \rightleftarrows \mathbf{Fig}$.

Using this classification, we say that an utterance u is a *placing* if it asserts that

$$ref(t_u) \models_{Fig_s} N_u.$$

Again, the difference between this placing and the corresponding stating is not one of truth, because $ref(t_u) \vDash_{Fig_s} N_u$ if and only if $ref(t_u) \vDash_{Fig} sense(N_u)$, but of what the speaker is doing.

Instancing

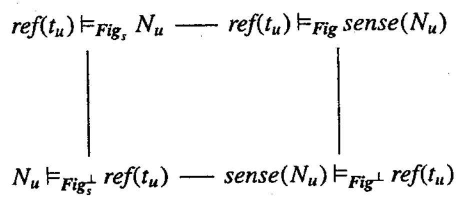
Given the resources at hand, one possibility is left open to us, namely, using the dual $Fig_s^\perp$ of the induced classification. We would hope that Austin's fourth type of assertion would assert

$$N_u \vDash_{Fig_s^\perp} ref(t_u).$$

Of the four, instancing is the speech act to which Austin pays least attention. He says merely that to instance is to cite t as an instance of N.

Is This What Austin Had in Mind?

Using the framework of classifications and infomorphisms, we have come up with four possible things one can do with a sentence of the form "t is a N." Let us summarize these by means of the following diagram:



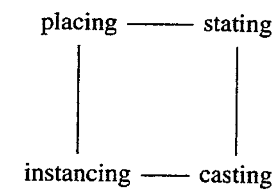
Each corner of the square represents a distinct possible content for an utterance u of

$$t \text{ is a } N,$$

depending on just what the speaker is doing with the utterance, that is, depending on its type as a speech act. The four contents are true or false together, but informationally they represent four distinct claims.

On the left side of this diagram, we are dealing with the classification Fig_s and its dual, relating items in the world and nouns in the language. On the right side, we are dealing with the classification Fig relating items in the world with their types. The top row of the diagram has to do with classifying items, either by nouns or by their senses. The bottom row has to do with finding items that are examples of nouns or their senses. Our diagram is put forward as an

explication of the following diagram from Austin:

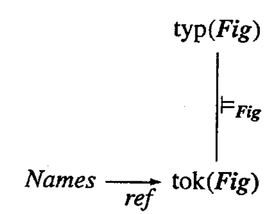


If this analysis is correct, it seems we could say, in a bit clearer way, just what the four speech acts are:

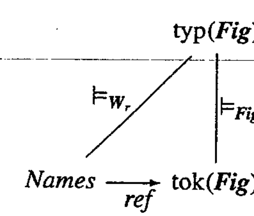
- In placing, one says what an object is called.
- In stating, one says what an object is.
- In instancing, one gives an example to which a given noun applies.
- In casting, one gives an example of a given type.

Let us stress once more that all four speech acts have equivalent truth conditions; where they differ is in what things they are classifying and what they are classifying these things with.

It is not clear from Austin's account why he treats names and noun phrases so asymmetrically. We could use the diagram



to induce a different classification W_r, namely, of names by the types of Fig.



If we use this classification, two more possibilities arise, corresponding to a different kind of information; information about the reference of names. Perhaps Austin used numerals for names just to avoid adding this sort of complication to his already complicated picture. But surely there are times when this is just what one is after. One is thus tempted to add two more assertive speech acts, let us call them *identifying* and *naming*.

- In identifying, one tells what kind of thing a name denotes: $t \models_{W_r} sense(N)$.
- In naming, one gives a name of something of a given kind: $sense(N) \models_{W_{r\perp}} t$.

There is yet another induced classification C, namely, of names by nouns, given by $t \models_C N$ if and only if $ref(t) \models_{Fig} sense(N)$. This brings up two more theoretical possibilities. Here the world does not really enter at all, except by the back door, so it is hard to see the point of using language in the two ways suggested by this classification and its dual. But with language use, it seems that anything that is possible is realized somewhere, so it would not surprise us to find real examples of this sort.

What are Propositions?

We have not used the notion of a proposition in this book. There are many different things one might mean by a proposition. If one wants to develop a theory of speech act contents, the above analysis suggests that one model a proposition as a triple $p = \langle A, a, \alpha \rangle$, where A is a classification, $a \in \text{tok}(A)$, and $\alpha \in \text{typ}(A)$. Then p would be *true* if $a \models_A \alpha$. What we have seen in this lecture is that, with this understanding of a proposition, there can be many distinct propositions associated with an utterance u, propositions that are linked to one another in systematic ways and have the same truth conditions. Which of these propositions should be assigned to u depends on us speech act type. This is far from a worked out theory of speech acts and how they would fit in a theory of information flow. But it shows, at least, that taking classifications seriously gives one the tools to make some distinctions that need to be developed for such a theory.

Lecture 18

Vagueness

A standard objection to classical logic has been its failure to come to grips with vague predicates and their associated problems and paradoxes. An analysis of the vague predicates "low," "medium,"and "high" (as applied to brightness of light bulbs) was implicit in Lecture 3. In this lecture we want to make the idea behind this treatment more explicit, thereby suggesting an information-theoretic line of research into vagueness. At best, this line of development would allow the information-flow perspective to contribute to the study of vagueness. At the very least, it should show that vagueness is not an insurmountable problem to the perspective offered in this book.

In this lecture we explore a different family of related vague predicates, "short," "medium," "tall," "taller," and "same height as." This family is simple enough to treat in some detail but complicated enough to exhibit three problems that are typical of vague predicates.

Information Flow Between Perspectives

The first problem is that different people, with differing circumstances, often have different standards in regard to what counts as being short or tall. In spite of the lack of any absolute standard, though, information flow is possible between people using these predicates. If Jane informs me that Mary is of medium height while she, Jane, is short, and if I consider Jane to be tall, then I know that I would consider Mary as tall as well. How is such reliable information flow possible between people with quite different standards of what counts as being tall?

The Logic of Vague Predicates

The second problem takes place within a fixed perspective. A given person may use vague predicates and know that certain cases are indeterminate. For a given person on the tallish side, Judith may not choose to decide whether that person is tall or of medium height. Thus she is not willing to grant that everyone is either short, medium, or tall. Thus it might seem that it would be hard to reason with vague predicates, but this does not seem to be the case. Judith, for example, maintains that the following are unproblematic:

If x is short then x is not tall.
If x is tall and y is taller than x then y is tall.
If x is of medium height and y is tall then y is taller than x.
If x is taller than y and y is taller than z then x is taller than z.

How can one give a principled semantic account of vague predicates that respects these intuitions?

The Sorites Paradox

The third problem, known as the sorites paradox, goes more directly to the heart of vagueness. One version of it, having to do with our predicates, runs as follows. Given the physical limits of human perceptual abilities, there is some positive number ϵ so that if the heights of two people x and y differ by less than ϵ, then x and y will of necessity be judged to be the same height by any accurate human observer. But now consider Billy, who was short but grew tall over the last year. Divide this year up into intervals $t_1 < \cdots < t_N$, where t_1 is the start of the year and t_N its end and where the intervals are chosen so that Billy's growth from t_i to t_{i+1} was less than ϵ. Then Billy would of necessity be judged to be the same height at t_{i+1} as at t_i for each $i < N$. But surely if he is short at t_i and the same height at t_{i+1} he is short at t_{i+1}. But then by induction it follows that Billy is short at each t_i and so is short at the end of the year.

18.1 Height Classifications

In this section, we propose that the vague predicates under discussion have a family of reasonable classifications, what we call "height" classifications. Within this framework, we will address the three problems raised above.

In order to be able to express interesting constraints, let Σ consist of the propositional formulas built up from

$$\text{SHORT(X), MEDIUM(X), TALL(X), TALLER(X, Y), SAMEHT(X, Y)},$$

using the usual propositional operators, where X, Y, ... are variables in some set *Var* of variables. Let B be some fixed set of instantaneous physical objects. We will usually be interested in the case where B is finite. Let $A = B^{Var}$ consist of all variable assignments taking values in B. We are interested in classifications A with $\text{typ}(A) = \Sigma$ and $\text{tok}(A) = A$. We will call such a classification *finite* if the set B is finite.[1] Intuitively,

$$a \models_A \text{TALL(X)} \wedge \text{TALLER(Y, X)}$$

if and only if the object $a(\text{X})$ assigned to X is classified as tall in A but the object $a(\text{Y})$ is even taller.

Let ***Ht*** be a state space that has as tokens the elements of B. The states of ***Ht*** consist of all nonnegative real numbers. We write $ht(b) = \text{state}_{Ht}(b)$ and call $ht(b)$ the *height* of b. We take this state space to model the heights of the objects in B using some standard unit of measurement.

Let $S = Ht^{Var}$. In other words, S is the product $\prod_{i \in I} Ht_i$, where $I = Var$ is the set of variables and $H_i = Ht$ for every i. Thus the tokens of S are functions from *Var* into B. This is the set of variable assignments taking values in B, that is, elements of A. The states are variable assignments taking values in the nonnegative real numbers.

We define what we mean by a height classification in terms of the notion of a "(height) regimentation," the idea being that any reasonable classification using these predicates must be compatible with one or more regimentations.

Definition 18.1. A *(height) regimentation* consists of a 4-tuple $r = \langle \epsilon, I_s, I_m, I_t \rangle$ satisfying the following conditions:

1. $\epsilon \geq 0$ is a real number in the set I_s, called the *tolerance* of r;
2. I_s, I_m, and I_t are mutually disjoint intervals of nonnegative real numbers such that I_t is closed upward, and I_s is closed downward;
3. if $r_t \in I_t$, $r_m \in I_m$, and $r_s \in I_s$, then $r_t - r_m \geq \epsilon$ and $r_m - r_s \geq \epsilon$.

The intervals I_s, I_m, and I_t of the regimentation r are the heights that are considered short, medium, and tall, respectively, under this regimentation. The tolerance of r is the least amount such that if $ht(b_1)$ is more than ϵ greater than $ht(b_2)$, then b_1 is considered taller than b_2. We allow for the possibility that $\epsilon = 0$, but we do not insist on it. The following result is obvious.

Proposition 18.2. *For any regimentation r, there is a unique token-identical contravariant pair of maps $f_r : A \rightleftarrows \text{Evt}(S)$ satisfying the following condition. (Here σ ranges over the states of S.)*

[1] Although Barwise is assumes responsibility for the explorations in Part III, he acknowledges Seligman's help in stressing the importance of finite classifications in response to an earlier version of this lecture.

1. $f_r(\text{SHORT(X)}) = \{\sigma \mid \sigma(\text{X}) \in I_s\}$
2. $f_r(\text{MEDIUM(X)}) = \{\sigma \mid \sigma(\text{X}) \in I_m\}$
3. $f_r(\text{TALL(X)}) = \{\sigma \mid \sigma(\text{X}) \in I_t\}$
4. $f_r(\text{TALLER(X,Y)}) = \{\sigma \mid \sigma(\text{X}) - \sigma(\text{Y}) > \epsilon\}$
5. $f_r(\text{SAMEHT(X,Y)}) = \{\sigma \mid |\sigma(\text{X}) - \sigma(\text{Y})| \leq \epsilon\}$
6. $f_r(\varphi \wedge \psi) = f_r(\varphi) \cap f_r(\psi)$
7. $f_r(\varphi \vee \psi) = f_r(\varphi) \cup f_r(\psi)$
8. $f_r(\neg\varphi) = -f_r(\varphi)$

Moreover, distinct regimentations give rise to distinct contravariant pairs.

Definition 18.3. A classification A with types and tokens as above is a *height classification* if there exists a regimentation r such that $f_r : A \rightleftarrows \text{Evt}(S)$ is an infomorphism. In this case, we say that r is a *regimentation of* A.

We suggest that height classifications and their associated logics are good models of the way we use the vague predicates in question.

The first thing to notice is that a height classification classifies things according to their heights in the following precise sense. For variable assignments $a_1, a_2 \in A$, define $a_1 \equiv a_2$ if for each variable X, $ht(a_1(\text{X})) = ht(a_2(\text{X}))$.

Proposition 18.4. *If $a_1 \equiv a_2$, then a_1 and a_2 are indistinguishable in every height classification. If a is an assignment assigning individuals of the same height to the variables* X *and* Y, *then*

$$a \models_A \text{SHORT(X)} \quad \textit{iff} \quad a \models_A \text{SHORT(Y)}$$

for every height classification A, and with parallel biconditionals for the other predicates.

In order for A to be a height classification, it must have at least one regimentation. But importantly, it will typically have many different regimentations. For example, we have the following proposition.

Proposition 18.5. *Let A be a height classification. If A is finite, then A has uncountably many distinct regimentations.*

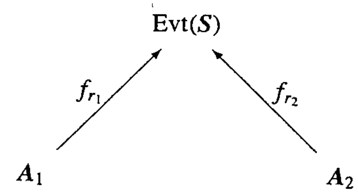
Proof. Given an interval I of real numbers and finitely many distinct real numbers $h_1, \ldots, h_n$, there are uncountably many intervals I such that $h_i \in I$ if and only if $h_i \in I'$, for each $i = 1, \ldots, n$. From this it follows that given any regimentation r, we can adjust the lower endpoint of the upper range of I_t in uncountably many ways and end up with the same classification. □

This result shows that we do not need to assume a specific regimentation in order to have a height classification. Rather, we can think of there being a nonempty class of regimentations implicit in any height classification. The same proof shows the following corollary.

Corollary 18.6. *Every finite height classification has a regimentation with tolerance $\epsilon > 0$.*

18.2 Information Flow

We can now address the first problem mentioned, that of information flow between different height classifications. Suppose A_1 and A_2 are both height classifications over our fixed set B of tokens. Although A_1 and A_2 may well show disagreement on how different objects are classified, there will typically be information flow between them because they are related by a channel as depicted below. Here r_1 and r_2 are any regimentations of A_1 and A_2.



Just exactly what information flow exists between A_1 and A_2 will depend on the properties of the channel, hence on what regimentations are compatible with each of the classifications. But no matter what regimentations are used it is easy to see the following (where we write A → B for ¬A ∨ B as usual):

$$f_1(\text{SHORT(X)} \wedge \text{MEDIUM(Y)}) \vdash_{\text{Log}(S)} f_2(\text{TALL(X)} \rightarrow \text{TALL(Y)}).$$

Consequently, $a \models_{A_1} \text{SHORT(X)} \wedge \text{MEDIUM(Y)}$ carries the information that $a \models_{A_2} \text{TALL(X)} \rightarrow \text{TALL(Y)}$. Hence

$$a \models_{A_1} \text{SHORT(X)} \wedge \text{MEDIUM(Y)} \quad \text{and} \quad a \models_{A_2} \text{TALL(X)}$$

carry the information that

$$a \models_{A_2} \text{TALL(Y)}.$$

This is just one of infinitely many examples that could be given.

18.3 An Intensional Logic

What logic should we consider to be "given" with the height classification A? One possibility, of course, is to take Log(A); this is sound and complete. This logic is a kind of extensionally given logic, extensional in that its constraints are entirely determined by the extensions of the types of A, regardless of what they mean.

A second possibility, one that is more intensional, is to take $\text{Log}^\circ_r(A) = f_r^{-1}[\text{Log}(S)]$ for some particular regimentation r. By calling $\text{Log}^\circ_r(A)$ "intensional," we mean that the constraints of $\text{Log}^\circ_r(A)$ are determined by the meaning of the types rather than by what their extension in A happens to be. This logic is also sound, being the inverse image of a sound logic under a token surjective infomorphism.

As we have seen, however, there are typically infinitely many different regimentations compatible with a given height classification. A particular regimentation r may well have properties that result in some constraints holding in $\text{Log}^\circ_r(A)$ that would not hold in some other $\text{Log}^\circ_{r'}(A)$. A more canonical choice is to take the meet (i.e., greatest lower bound) of these logics.

Definition 18.7. Let A be a height classification. The extensional height logic of A is just Log(A). The *intensional logic* of A is

$$\text{Log}^\circ(A) = \bigsqcap \{\text{Log}^\circ_r(A) \mid r \text{ a regimentation of } A\}.$$

We write $\vdash^\circ_A$ for the consequence relation of $\text{Log}^\circ(A)$. Being the meet of sound logics, this logic is also sound.

Proposition 18.8. *For any height classification A, the following are constraints of* $\text{Log}^\circ(A)$. *(Because* $\text{Log}^\circ(A) \sqsubseteq \text{Log}(A)$, *they are* a fortiori *constraints of the latter.)*

$$\text{SHORT(X)} \vdash^\circ_A \neg\text{TALL(X)}$$

$$\text{TALL(X), TALLER(Y, X)} \vdash^\circ_A \text{TALL(Y)}$$

$$\text{MEDIUM(X), TALL(Y)} \vdash^\circ_A \text{TALLER(Y, X)}$$

$$\text{TALLER(X, Y)} \wedge \text{TALLER(Y, Z)} \vdash^\circ_A \text{TALLER(X, Z)}$$

Proof. Each of these is easily verified to hold in each $\text{Log}^\circ_r(A)$, hence in their meet $\text{Log}^\circ(A)$. □

The constraints listed in the above propositions are, of course, just a small, representative sample. More interestingly, let us turn to some of the differences

between the intensional and extensional logics. We call a height classification A *determinate* if each object b is classified as one of short, medium, or tall in A.

Proposition 18.9. *Let A be a height classification.*

1. If A is determinate, then

$$\vdash_A \text{SHORT(X)} \vee \text{MEDIUM(X)} \vee \text{TALL(X)}.$$

2. If A is finite, then

$$\nvdash^\circ_A \text{SHORT(X)} \vee \text{MEDIUM(X)} \vee \text{TALL(X)}.$$

Proof. Here (1) is practically a restatement of the definition of determinate. For (2), we note that if A is finite, then it will always have a regimentation r of A where the intervals of the regimentation do not exhaust the nonnegative real numbers. Then $f_r(\text{SHORT(X)} \vee \text{MEDIUM(X)} \vee \text{TALL(X)})$ does not exhaust the set of states so

$$\nvdash_{\text{Log}^\circ_r(A)} \text{SHORT(X)} \vee \text{MEDIUM(X)} \vee \text{TALL(X)}.$$

But $\text{Log}^\circ(A) \sqsubseteq \text{Log}^\circ_r(A)$, so the result follows. □

Corollary 18.10. *If A is finite and determinate, then the logic* $\text{Log}^\circ(A)$ *is strictly weaker than the logic* Log(A).

This proposition gives a rigorous form to the intuition that even if it happens to be the case that everything we are classifying is clearly one of the three sizes, short, medium, or tall, there could have been borderline cases too close to call.

18.4 The Sorites Paradox

We now turn to the third and final problem about vague predicates raised earlier, the sorites paradox. We start with a divergence between the intensional and extensional logics that is clearly relevant to this paradox. Call a classification A *precise* if $a \models_A \text{SAMEHT(X,Y)}$ implies $ht(a(\text{X})) = ht(a(\text{Y}))$, for all $a \in A$. (The converse is automatic.) It is clear that A is precise if it has a regimentation with zero tolerance.

Proposition 18.11. *Let A be a height assignment.*

1. *If A is precise, then*

$$\text{SAMEHT}(X, Y) \wedge \text{SAMEHT}(Y, Z) \vdash_A \text{SAMEHT}(X, Z).$$

2. *If A is finite, then*

$$\text{SAMEHT}(X, Y) \wedge \text{SAMEHT}(Y, Z) \nvdash^{\circ}_{A} \text{SAMEHT}(X, Z).$$

Proof. The proof of (1) is clear. To prove (2), note that because A is finite, it has a regimentation r with tolerance $\epsilon > 0$ by Corollary 18.6. Let σ be any state with $\sigma(Y) = \sigma(X) + .6\epsilon$ and $\sigma(Z) = \sigma(Y) + .6\epsilon$. Such states will be in f_r of the left-hand side but not in f_r of the right-hand side because $\sigma(Z) - \sigma(X) = 1.2\epsilon > \epsilon$. □

The relationship to the sorites paradox should be evident. The intensional logic does not provide us with the constraint telling us that being the same height is transitive, even if our classification happens to be precise. (Notice that there will be many imprecise height classifications A that also have this constraint in their extensional logic.) Let us make the connection with the paradox more transparent by means of the following result.

Theorem 18.12. *Let A be a height classification.*

1. *If A is precise, then for all integers* $N \geq 2$:

$$\text{SHORT}(X_1) \wedge \text{SAMEHT}(X_1, X_2) \wedge \cdots \wedge \text{SAMEHT}(X_{N-1}, X_N) \vdash_A \text{SHORT}(X_N),$$

and so

$$\text{SHORT}(X_1) \wedge \text{SAMEHT}(X_1, X_2) \wedge \cdots \wedge \text{SAMEHT}(X_{N-1}, X_N) \vdash_A \neg\text{TALL}(X_N).$$

2. *Whether or not A is precise, if it is finite then for sufficiently large integers N,*

$$\text{SHORT}(X_1) \wedge \text{SAMEHT}(X_1, X_2) \wedge \cdots \wedge \text{SAMEHT}(X_{N-1}, X_N) \nvdash^{\circ}_{A} \neg\text{TALL}(X_N)$$

Proof. Again, the first statement is trivial. For the second, pick a regimentation r of A with tolerance $\epsilon > 0$ and I_t nonempty. Let K be the least integer such that $K\epsilon/2 \in I_t$; such a K must exist because I_t is nonempty and closed upward. Let σ be any state satisfying the following: $\sigma(x_1) = \epsilon/2$ and $\sigma(X_{i+1}) = \sigma(X_i) + \epsilon/2$ for every i. Then

$$\sigma(X_N) = N\epsilon/2 \geq K\epsilon/2$$

for every $N \geq K$. Hence $\sigma(X_N) \in I_t$ because $K\epsilon/2 \in I_t$ and I_t is closed upward. Consequently, σ is in the translation via f_r of the type on the left of the sequent but not of the type on the right. □

Definition 18.13. Let A be a height classification. The *sorites number* of a regimentation r is the least integer N such that

$$\text{SHORT}(X_1) \wedge \text{SAMEHT}(X_1, X_2) \wedge \cdots \wedge \text{SAMEHT}(X_{N-1}, X_N) \wedge \text{TALL}(X_N)$$

is consistent in $\text{Log}^{\circ}_{r}(A)$, if it exists. The sorites number of A is the least N such that the above is consistent in $\text{Log}^{\circ}(A)$, if it exists.

The theorem says every finite classification has a sorites number. (We show how to calculate this number in Exercise 18.5.) What this means is that it would be entirely possible to have a chain of individuals, $b_1, \ldots, b_N$, where N is the sorites number of A, such that b_1 is classified as short, each b_i is classified as being the same height as b_{i+1}, for $i < N$, and yet for b_N to be tall, even if this does not happen in the classification A.

Let us go back and analyze the argument in the sorites paradox. The first step is to assert there is a number $\epsilon > 0$ so that the sequent

$$\text{SHORT}(X) \wedge \text{SAMEHT}(X,Y) \vdash \text{SHORT}(Y)$$

holds of all tokens a with $|ht(a(X)) - ht(a(Y))| < \epsilon$. This is possible as long as the height classification A is finite (and for many infinite classifications as well). But the conclusion of the argument deals not just with this classification but with what would happen if we were to use the fixed regimentation and add N new tokens to the classification, where N is the sorites number of A. This would result in a different classification and in that classification the above sequent would fail to hold for many tokens a with $|ht(a(X)) - ht(a(Y))| < \epsilon$.

Non-Archimedian Regimentations

We have required that our regimentations live in the field of real numbers. An alternative suggestion would allow them to live in non-Archimedian fields, like the fields of nonstandard real numbers. One could then allow the tolerance to be infinitesimal, and sorites numbers could be infinite. This goes along with the discovery in recent years that nonstandard analysis often gives an elegant way to model the differences between the very small and the very large.

Exercises

18.1. Is it consistent with the framework presented here for a short office building be taller than a tall person?

The exercises below all assume the following setup. In a sixth grade gym class having twenty students, the coach has classified students by height before dividing them up into basketball teams so that she can put the same number of short, medium, and tall girls on each of the four teams. The shortest student is 3′8″, the tallest is 5′6″, and the smallest difference in heights between students of different heights is .25″. The coach classified all the girls less than four feet tall as short, those five feet or over as tall, and those in between as medium height. The coach used measuring tools that have a precision of .1″.

18.2. Assuming a unit of measure of 1″, describe a regimentation of A.

18.3. Is the classification determinate? Is it precise?

18.4. Give an upper bound for the sorites number of A.

18.5. Determine the sorites numbers exact value assuming that the shortest girl of medium height is 4′1″ and the tallest girl of medium height is 4′10.5″.

Lecture 19

Commonsense Reasoning

Among the problems that have beset the field of artificial intelligence, or AI, two have a particularly logical flavor. One is the problem of nonmonotonicity referred to in Part I. The other is the so-called frame problem. In this lecture, we suggest that ideas from the theory presented here, combined with ideas and techniques routinely used in state-space modeling in the sciences, suggest a new approach to these problems.

Nonmonotonicity

The rule of Weakening implies what is often called monotonicity:

If $\Gamma \vdash \Delta$ then $\Gamma, \alpha \vdash \Delta$.

The problem of nonmonotonicity, we recall, has to do with cases where one is disinclined to accept a constraint of the form $\Gamma, \alpha \vdash \Delta$ even though one accepts as a constraint $\Gamma \vdash \Delta$. The following is an example we will discuss in this lecture.

Example 19.1. Judith has a certain commonsense understanding of her home's heating system – the furnace, thermostat, vents, and the way they function to keep her house warm. Her understanding gives rise to inferences like the following.

(α_1) The thermostat is set between sixty-five and seventy degrees.
(α_2) The room temperature is fifty-eight degrees.
$\vdash$ (β) Hot air is coming out of the vents.

It seems that $\alpha_1, \alpha_2 \vdash \beta$ is a constraint that Judith uses quite regularly and unproblematically in reasoning about her heating system. However, during a recent blizzard she was forced to add the premise

(α_3) The power is off.

Monotonicity (Weakening) seems to fail, because surely

$$\alpha_1, \alpha_2, \alpha_3 \nvdash \beta.$$

In fact, we would expect Judith to realize that

$$\alpha_1, \alpha_2, \alpha_3 \vdash \neg\beta.$$

The problem of nonmonotonicity has been a major area of research in AI since the late 1970s. Because nonmonotonicity clearly violates the classically valid law of Weakening, a property of all local logics, the problem might be seen as a major one for our account. It seems, though, to be closely related to a problem we have already addressed in this book, that of accounting for exceptions to lawlike regularities. This suggests that implicit in our account of exceptionality is some sort of proposal about how to solve the problem of nonmonotonicity. In this lecture, we propose a solution that involves shifting between local logics, which is, as we have seen, equivalent to shifting between information channels.

The Frame Problem

The "frame" problem has to do with inferring the consequences of a change in a system. We can get a feeling for the problem by means of an example.

Example 19.2. Suppose the temperature at Judith's thermostat is seventy-two degrees and the thermostat is set at sixty-eight degrees. Thus there is no hot air coming out. Now the temperature drops to sixty-five degrees. We want to infer that the furnace comes on and hot air comes out of the vents. The difficulty in obtaining this inference stems from the fact that one change (the temperature dropping) produces other changes, like the furnace coming on, but leaves many other things unaffected. For example, the temperature dropping does not cause the thermostat setting to drop. If it did, the heat would not come on after all.

People are reasonably good at inferring the immediate consequences of basic actions, what other things will change, and how, and what will remain unaffected. In trying to model this sort of inference using axiomatic theories and logical inference, AI researchers have found it very difficult to mimic this performance. The problem is not so much in stating what other changes follow from some basic action, like a drop in the temperature. The difficulty is that it is not feasible to state an explicit axiom that says that a drop in the room's temperature does not cause the furnace to explode or the stock market to crash, and so on.

19.1 The Dimension of a State Space

In Lecture 1, we reviewed the standard scientific practice of treating the state of any system as determined by the values taken on by various attributes, the so-called observables of the system. These values are typically taken to be measurable quantities and so take on values in the set $\mathbb{R}$ of real numbers. Consequently, the total state of the system is modeled by a vector $\sigma \in \mathbb{R}^n$ for some n, the "dimension" of the state space. We saw in Lecture 8 that projections, the natural morphisms that go with state spaces, are covariant and that, consequently, products play the information-theoretic role in state spaces that sums play in classifications. This shows that the traditional use of product spaces as the set of possible states of a system goes with the turf – it is not an accident of history.

Definition 19.3. A *real-valued state space* is a state space S such that $\Omega \subseteq \mathbb{R}^n$, for some natural number n, called the *dimension* of the space.[1] The set $n = \{0, 1, \ldots, n-1\}$ is called the set of *observables* and the projection function $\pi_i(\sigma) = \sigma_i$, the ith coordinate of σ, is called the ith *observation function* of the system. If σ is the state of s, then σ_i is called the value of the ith observable on s. We assume that the set of observables is partitioned into two sets:

$$Observables = J \cup O,$$

where J is the set of input observables and O is the set of output observables. Each output observable $o \in O$ is assumed to be of the form

$$\sigma_o = F_o(\vec{\sigma}_J)$$

for some function F_o of the input observables $\vec{\sigma}_J$.

We are taught to think of the world as being three or four dimensional, but scientific practice shows this to be a vast oversimplification. The spatial-temporal

[1] An interesting and useful introduction to real-valued state spaces can be found in the first chapter of Casti (1992). It is worth distinguishing two decisions implicit in the definition of a real-valued state space. One is to be cognizant of the product structure of the set of states of the system. The other is to restrict to those observables whose values are determined by magnitude, and so nicely modeled by real numbers. For the purposes of this lecture, the first decision is the important one. At the cost of being a bit more wordy, we could generalize everything we do here to a setting where we allowed each observable i to take values in a set V_i. This would probably be important in areas like linguistics where the structure of the state has much more to it than mere magnitude.

location of an object is a region in four-space, but its total state typically has more degrees of freedom. We saw, for example, that the Newtonian state of a system consisting of n bodies is classically modeled by a state space of dimension $6n$. In Lecture 3, we modeled the light circuit with a four-dimensional state space, completely setting aside issues of time and space. We then modeled actions on the light circuit with an eight-dimensional state space. Here we suggest ways of exploiting the dimensionality of state spaces for addressing nonmonotonicity and the frame problem.

Example 19.4. We use Judith's heating system as a running example and so start by building a real-valued state space S_{hs} of dimension 7 for describing this system. For tokens, take some set of objects without additional mathematical structure. Intuitively, these are to be instances of Judith's complete heating system at various times, including the vents, the thermostat, the furnace, the ambient air in the room where the thermostat is located, and so forth. We assume that each state is determined by a combination of the following seven "observables" of the system:

Thermostat setting: some real σ_1 between 55 and 80;
Room temperature: (in Fahrenheit) a real σ_2 between 20 and 110;
Power: $\sigma_3 = 1$ (on) or 0 (off);
Exhaust vents: $\sigma_4 = 0$ (blocked) or 1 (clear);
Operating condition: $\sigma_5 = -1$ (cooling), 0 (off), or 1 (heating);
Running: $\sigma_6 = 1$ (on) or 0 (not on);
Output air temperature: a real σ_7 between 20 and 110.

Thus for states we let Ω be the set of vectors $B = \langle \sigma_1, \ldots, \sigma_7 \rangle \in \mathbb{R}^7$. We take $\sigma_1, \ldots, \sigma_5$ as inputs and σ_6 and σ_7 as outputs. We restrict the states to those satisfying the following equations:

$$\sigma_6 = pos(\sigma_5 \cdot sg(\sigma_1 - \sigma_2)) \cdot \sigma_3 \cdot \sigma_4$$

$$\sigma_7 = \begin{cases} 55 & \text{if } \sigma_5 \cdot \sigma_6 = -1 \\ 80 & \text{if } \sigma_5 \cdot \sigma_6 = +1 \\ \sigma_2 & \text{otherwise,} \end{cases}$$

where

$$sg(r) = \begin{cases} +1 & \text{if } r \geq 2 \\ 0 & \text{if } |r| < 2 \\ -1 & \text{if } r \leq -2 \end{cases}$$

$$pos(r) = \begin{cases} 1 & \text{if } r > 0 \\ 0 & \text{if } r \leq 0. \end{cases}$$

We let S_{hs} consist of these tokens and states, with some total function *state* mapping the tokens into the states.

The types used in our nonmonotonicity example are represented in $\text{Evt}(S_{hs})$ by

$$\begin{aligned} \alpha_1 &= \{\sigma \in \Omega_{hs} \mid 65 \leq \sigma_1 \leq 70\} \\ \alpha_2 &= \{\sigma \in \Omega_{hs} \mid \sigma_2 = 58\} \\ \alpha_3 &= \{\sigma \in \Omega_{hs} \mid \sigma_3 = 0\} \\ \beta &= \{\sigma \in \Omega_{hs} \mid \sigma_6 = 1 \text{ and } \sigma_7 > \sigma_2\}. \end{aligned}$$

It is usually assumed that the input observables are *independent* in that the observer can vary the inputs independently. This amounts to the following requirement: if σ is some state, $i \in J$ is some input observable, and r is a value of σ'_i for some state σ', then there is a state σ'' such that $\sigma''_i = r$ and $\sigma''_j = \sigma_j$ for all $j \neq i$. One might think of this as a precondition as to what it would mean for an observable to be an input to the system.

This assumption is clearly related to the frame problem. When we make a change to a system, we typically change one of the inputs, with concommitant changes in output observables; we do not expect a change in one input to produce a change in any other inputs. We will make this idea more explicit later in the lecture. We will not assume that the input observables are independent, preferring to state explicitly where the assumption plays a role in our proposals.

When working with a state-space model, it is customary to partition the inputs into two, $J = I \cup P$. The input observables in I are called the *explicit inputs* of the system and those in P the *parameters* of the system. Intuitively, the parameters are those inputs that are held fixed in any given computation or discussion. We will not make a permanent division into explicit inputs and parameters, but will instead build it into our notion of a background condition in the next section.

Example 19.5. In our heating system example S_{hs}, it is natural to take to take σ_1, σ_2 as explicit inputs, and σ_3, σ_4, and σ_5 as parameters. The inputs are clearly independent.

19.2 Nonmonotonicity

The proposal made here was inspired by the following claim:

"...error or surprise always involves a discrepancy between the objects [tokens] *open* to interaction and the abstractions [states] *closed* to those same interactions. In principle, the remedy for closing this gap is equally clear: augment the description by including more observables to account for the unmodeled interactions (Casti, 1992, p. 25)."