The Logical Form of Status-Function Declarations[*]

Richard Evans
THE SIMS DIVISION Maxis/Electronic Arts
richarde@ea.com

ABSTRACT
We are able to participate in countless different sorts of social practice. This indefinite set of capacities must be explainable in terms of a finite stock of capacities. This paper compares and contrasts two different explanations. A standard decomposition of the capacity to participate in social practices goes something like this: the interpreter arrives on the scene with a stock of generic practice-types. He looks at the current scene to fill-in the current tokens of these types. He looks at the current state of these practice tokens to see what actions are available to him. He uses his current desires to choose between these various possible actions. I argue that this standard explanation is defective, drawing on arguments by Searle and Wittgenstein and Garfinkel. I propose an alternative explanation, in which the participants must continually show each other the state of the scene in order to maintain the scene's intelligibility. I provide a simple formal language in which to describe this alternative approach, in which we can state quite precisely what someone is (and should be) doing when they participate in a practice. This language is related to both deontic and epistemic logics, but it is much simpler – it does not include the classic propositional connectives, and it is driven by a very different set of assumptions. The inspirations for this formal language are Searle's analysis of directions of fit, Wittgenstein's remarks on rule-following and Garfinkel's ethnomethodology.

1. *Decomposing the Capacity to Participate in Practices*

As social creatures, we are capable of participating in an indefinite variety of social practices. As Wittgenstein reminds us: "But how many kinds of sentence are there? Say assertion, question, and command? – There are countless kinds: countless different kinds of use of what we call 'symbols', 'words', 'sentences'. And this multiplicity is not something fixed, given once for all; but new types of language, new language- games, as we may say, come into existence, and others become obsolete and get forgotten." [*Philosophical Investigations* §23]

How do we manage to participate in so many different kinds of social practice? We cannot have a countless number of primitive capacities. There

---

must be a small finite stock of capacities which can be recursively combined to produce our ability to understand and participate in an indefinite number of different types of practice.

Consider, for example, a mechanical piano, which can play any number of tunes. These different capacities – the capacity to play Humpty Dumpty, the capacity to play Yankee Doodle, etc – cannot be separate primitive capacities. There are just too many of them. There must be a general explanation which can say how these infinitely many capacities can come from a small finite collection of general capacities. In this case, the explanation is based on the mechanical piano's ability to read any piano roll (a roll of paper with perforations representing notes). Each tune can be represented by a particular piano roll, so the capacity to play an indefinite number of tunes can be explained in terms of one general capacity (to play any piano roll) plus a mapping from each of the different tunes to its representation on a piano roll.

This sort of example suggests one obvious way to break-down the capacity to participate into a small set of finite components: invoking a distinction between practice-types and tokens.

2. *The Type/Token Approach*

There is one seemingly natural way to explain or decompose the ability to participate in practices. This is what I will call the *Type/Token* approach.
It goes like this: our training and history has given us a library of practice-types which we understand. In any particular situation, we apply these types to the current situation, instantiating particular tokens of those types.

Now we can be in many different practices simultaneously. (Think of a mother who is standing in line while trying to restrain her infant). So the current social scene is the union of the practice tokens we are currently in. Each practice token issues us with new options, actions we are capable of doing: new capacities. We use our desire to weigh these various actions, to decide what to do.

In this picture, participating in a practice requires two primitive mental capacities: Scorekeeping and Revelation. Scorekeeping is the ability to use facts about the current situation to turn practice-types into practice-tokens: particular instances of practices, instantiated with particular individual agents and objects (for example, this *particular* chess game between Jack and Jill). Revelation is the ability to understand, based on past history, a new

practice-type. (This might be, for instance, by using Bayesian statistical learning from past examples).

In this picture, we see the particular practice instances of the current social scene through the lens of the practice-types.

It is helpful to focus on a concrete case. Throughout this paper, I shall use the example of a queue: people standing in line, waiting to get on the bus. According to the Type/Token approach, someone who is participating in the bus queue is doing so because he has an understanding of a practice-type – queuing – and he has used information about the particular scene around him to instantiate a particular practice-token – queuing for this particular bus with these particular people on this particular day in February. A scorekeeper participating in this queue-instance sees that the queue makes a new action available to him: he is able to queue up. He evaluates this new action by assigning it a score, based on his desires, and compares it with what he is currently doing. How did he come to an understanding of the general practice-type of queuing? According to the Type/Token approach being considered here, his capacity for Revelation made the practice-type available to him, perhaps as a Bayesian statistical generalization from observing various examples of queuing behavior in the past.

History

↓

Revelation

↓

Practice Types   Current Facts

Scorekeeping

↓

Practice Tokens

↓

Action Extraction

↓

Choices   Desires

Humean decision making

↓

Chosen action
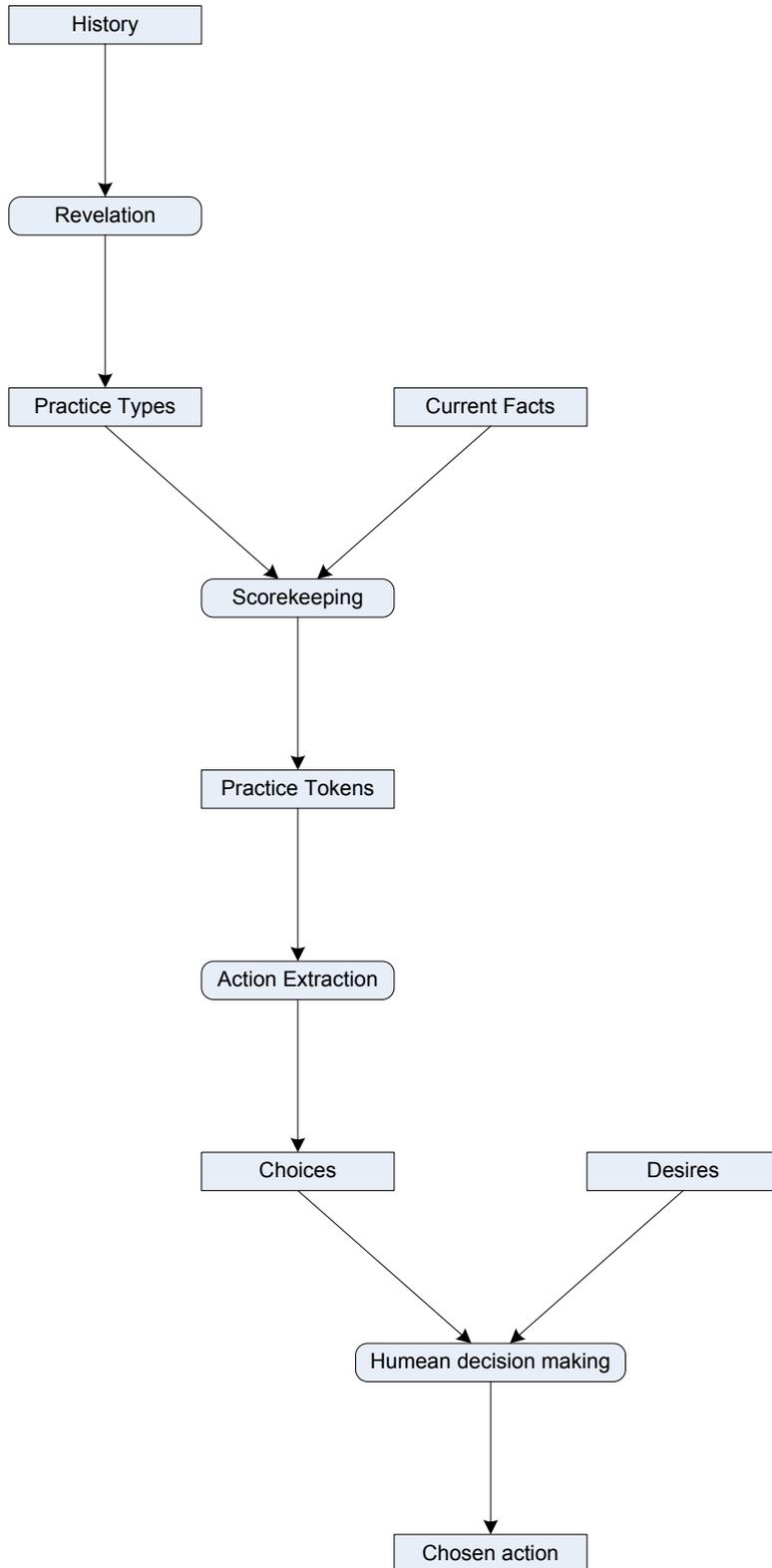
Here, processes are represented by curved boxes, data by rectangles, and information flow by arrows. According to the Type/Token approach, we

construct practice-types from past data using a faculty called Revelation. We use Scorekeeping to instantiate practice-types into practice-tokens, using facts about the current situation. We extract from the practice-tokens the actions that are currently available to us. We weigh up the various choices using our desires to choose the action which best satisfies our desires.

I used to subscribe to the Type/Token approach – I didn't really understand there was an alternative. I used to think this is what Wittgenstein thought, too. I gave a talk, entitled "Implementing Wittgenstein", about this. I now think a better name for the talk would be "Completely Failing to Implement Wittgenstein".

I now think the Type/Token approach is wrong. The rest of this paper will show why it is wrong, and describe an alternative approach.

3. *Problems with the Type/Token Approach*

There are three main problems with the Type/Token approach to decomposing the capacity to participate in practices:

Firstly, the Type/Token approach assumes a fundamental mental capacity – scorekeeping – which is outside the practices which it seeks to understand: scorekeeping is just something our minds do in private. But scorekeeping isn't a private mental capacity. Keeping score on a practice, knowing what state it is in, is a public action. Others can see that we are doing it, and we are accountable for how we do it. *Scorekeeping is part of the very practice which it seeks to understand.*

Secondly, and even more fundamentally, it misrepresents the distinction between the interpreter, a newcomer who has to understand the practice, and the expert participants, who are at home in the practice, who are effortlessly participating. According to the Type/Token approach, the newcomer has to do the work of seeing the state of the practice just from watching the actions of the expert participants. This picture assumes a fundamental asymmetry of responsibility: the interpreter is responsible for scorekeeping, but accords no responsibility to the participants who are already involved. The truth is rather that all participants are continually responsible for *showing* or *manifesting* the state of the practice to each other. Showing the state of the practice is a public witnessable act, just like seeing the state of the practice, and is equally accountable - failures to show may be censored. *Showing the state of the practice is part of the practice which it seeks to describe.*

Thirdly, it assumes a Humean understanding of decision-making as a non-rational weighing of desires. But this irrational weighing puts a depressing

limit on our capacity for rational self-realization, and renders us merely beasts who calculate. Deciding what to do is itself a public, seeable act, subject to normative assessment. *Deciding what to do in a situation is part of the very situation it seeks to evaluate.*

I shall elaborate on each of these.

4 *Scorekeeping is part of the very practice which it seeks to understand*

Scorekeeping is not a hidden private mental act, but a public act: others can see that we are doing it.

Consider the people queuing for the bus. A newcomer arrives, and looks for the end of the line. His looking-for-the-end-of-the-line is itself something that others can see him doing. As Garfinkel remarks, "the end of the line is watchably searched-for". Garfinkel is here stressing that the seeing which the newcomers should do is a public seeing which can itself be seen by others. His scorekeeping is directly visible to the others. If he fails to do it correctly, if he fails to see the back of the queue correctly, he is culpable and will be corrected by the others. When people join queues they are expected to position themselves correctly. But it is not just his placement which is subject to normative assessment - his scorekeeping of other people's placement is equally susceptible.

Wittgenstein writes: "Try not to think of understanding as a 'mental process' at all. – For *that* is the expression which confuses you… In the sense in which there are processes (including mental processes) which are characteristic of understanding, under standing is not a mental process." [*Philosophical Investigations* §154]

Scorekeeping isn't something we do secretly in our private mental kingdom. Scorekeeping is a public action. Because it is public, scorekeeping is itself subject to normative assessment and evaluation: others can see that we are doing it, and can see when we are doing it *wrong*. Participants are expected to see that the practice is in a certain state. Others will notice if a participant has failed to see something that he should see, and may rebuke him because of his failure to see.

5. *Showing the state of the practice is part of the practice which it seeks to describe*

The Type/Token approach is often motivated by seeing the situation through the eyes of the newcomer, or outsider, who has to understand what the

participants are doing. We are immediately led to see the problem as the newcomer's problem: he alone has to figure out what is going on.

But it is not just the responsibility of the interpreter to see the state of the practice – it is equally the responsibility of the participants to continually *manifest* or *show* the state of the practice. In the Type/Token picture, we imagine the people who are at home in the practice participating effortlessly, and we imagine a newcomer, who wants to understand and join in. We ask – what does the newcomer have to do to understand and join in? But this question presupposes an *asymmetry of responsibility* between the newcomer and the regulars. In the alternative ethnomethodological perspective, the responsibility of creating intelligibility is *shared equally* amongst the participants. What it is to be a participant is for it to be the case that you should show the state of the practice.

The scene needs to be continually reinforced through our continually manifesting its state to each other, and this manifestation is our continual responsibility as participants.

Laurie Andersen, a performance artist, uses the following words in one of her pieces: "You're walking. And you don't always realize it, but you're always falling. With each step you fall forward slightly. And then catch yourself from falling. Over and over, you're falling. And then catching yourself from falling. And this is how you can be walking and falling at the same time. [*Walking & Falling*]"

Similarly, when we are in practices, intelligibility is continually being lost and found, and lost again and found again. *Showing* each other the state of the practice is what we do to pick ourselves up when we fall.

We are always continually showing each other the state of the situation we are in. This is not something we do just with children, or out of politeness – we have to continually *show* in order to be *in* the situation at all.

John Searle is well aware of this. In *Social Ontology: Some Basic Principles*, he uses status-function declarations as the explanatorily-fundamental action out of which social practices are constructed. These status-function declarations are precisely a way of showing the state of the situation. (Although, perhaps, from the ethnomethodological perspective being proposed here, Searle does not go quite far enough. He sometimes gives the impression that these showings are things you do *once*, at the beginning of the practice, to set up the roles - whereas the ethnomethodologist sees the showings as things which have to be *continually reinforced*. The creation of social intelligibility is an ongoing achievement which is in continual danger of falling apart, and needs to be supported with showings at every stage).

Queues are an excellent example of this. The participants are preoccupied with place-work: showing the state of the practice by their orientation and position. As Garfinkel says: "Persons are positioned and oriented to make a great positioning to-do about places in line."
They produce the intelligibility of the local queue by showing their place in line. In *Autochthonous Properties of Formatted Queues*, Garfinkel describes the work of the participants that is performed in order to achieve the coherence of the queue:

- Each person is showing he is behind the one in front.
- When something goes wrong, when a newcomer attempts to join the queue in the middle, perhaps because he hasn't noticed the people behind, the participants will show him that and how he has gone wrong. This showing is not just grumpiness or irritation on their part. This showing is part of the work of producing the intelligibility of the scene.
- "Consider how frequently mothers restrain their children" in queues – they are not doing this just out of politeness or some old-fashioned sense of social propriety – it is part of the work of being in the queue that they show their position: they are responsible for their children, who are also in the queue, so they also have to keep their position for them.
- Showing the state of the practice is the continual duty of the participants, without which the practice would cease to be. This work of showing what should be done is continual, relied upon, but unnoticed.

If this work of showing were to cease, the practice would cease to be. This fundamental ethnomethodological point is obscured by the philosopher's favorite example of a social practice – a board-game. In a board-game like chess, the board and the pieces themselves do the work of showing the state of the practice[1]. In this respect, philosopher's favorite example of a practice is seriously misleading: this case is an example of the *mechanization* of practice,

---

[1] Well, almost – there are a couple of aspects of the state of the scene which are not explicit in the board position: whose move it is, and how many times the same move sequence has been repeate

where the work of manifesting has been handed down from the participants to the objects[2].

The Type/Token approach assumes we see the local practice token through the lens of the practice-type. It assumes a sharp dichotomy between routine Scorekeeping and inspired Revelation. But this dichotomy is false to the facts.

Consider, for example, variations on the typical queue. Consider the scene that has sprung up around the coffee machine in the office: people have arrived to get coffee in a specific order; they themselves know the order they arrived in, but the local area around the coffee machine is cramped and not conducive to the formation of an actual line, so the participants are unable to show their position in line by their placement and orientation. This is what Garfinkel calls a "local interactional crush". Some aspects of the queue have been preserved, but others have been lost. Or consider a variant queue in which, instead of each participant standing behind the person in front of him, he has a laser pointer, and he points the laser's light at the person in front of him. Is this a queue? The Type/Token approach would either have to have a massively general version of a queue, or it would have to allow different sorts of queue types. And how does someone learn a new queue type? This is why I named the faculty "Revelation" - to underline its mysteriousness.

Wittgenstein was well aware of this: "We can easily imagine people amusing themselves in a field by playing with a ball so as to start various existing games, but playing many without finishing them and in between throwing the ball aimlessly into the air, chasing one another with the ball and bombarding one another for a joke and so on. And now someone says: The whole time they are playing a ball game and following definite rules at every throw. [*Philosophical Investigations*, §83]

Here he attacks the root idea that we see the local scene through the lens of a given set of practice-types.

The ethnomethodological/Wittgenstinian alternative has a different understanding of how we understand new situations. Not only are seeing and showing public accountable acts, but showing what others should see is also part of the practice. *Training (showing others what they should see) is also part of the practice which it seeks to teach others how to participate in.* Because the newcomer is shown how to see, he has a way into the practice which does not

---

[2] This is why Garfinkel urges us to consider cases of double-blind chess — cases where both players are blind-folded, and have to speak their moves to each other. Now that we no longer have the pieces to show us the state of the situation, the work needed to show the state of the scene comes immediately to the foreground.

involve revelation of a new practice-type. If he did, and if Bayesian learning or something similar is needed to learn a practice type, people would never be able to participate in new practices unless they had seen several prior examples. But the reality is that people just *immediately understand*. What actually happens is not that he performs induction over past behaviors to generate a new type, but that he is trained how to see this particular scene.

He understands the local scene immediately, *by letting others show him how to see*. In this alternative picture, practice-types are generalizations which are made *subsequent* to understanding the local scene we are in, not concepts which we need prior to understanding the local scene. In this alternative picture, generalizations are models of the practice[3] which *lose* information.

The showings and seeings themselves are lost when we generalize from the token to the type. In the alternative ethnomethodological picture, the pattern is new in every moment, and there is no sharp distinction between scorekeeping and revelation.

6. *The Humean understanding of Decision-Making is flawed*

We have looked at two fundamental problems with the Type/Token approach. The third problem is that it gives an inadequate conception of an agent's decision-making.

In some sense, participation in practices expands our possibilities and allows us to do things we couldn't do otherwise. This constitutive nature of practices can be understood in two ways:

- Participating in the practice gives us new (non-deontic) *capacities*
- Participating in the practice gives us new *norms* to follow

Let us consider a concrete example. Because I am playing chess, and because the current position is such, I am able to castle on the king's side. I wouldn't be able to castle on the king's side if I wasn't playing chess. Now is this expansion of my agency to be understood as a new capacity (like being able to fly), or should it be understood as a new norm (which can be followed or not followed). When deciding whether to castle, do I need to use *desires* to weigh up the relative benefits of castling and not-castling, or alternatively do I use *reason* to resolve the various conflicting norms?

---

[3] For this reason, ethnomethodologists typically use the word "scene" to represent the local norms making up the current moment, rather than the word "practice" which admits of a type/token distinction.

The type-token approach sees practices as expanding our agency by giving us new (non-deontic) capacities. In this approach, we need our decision-making (based on desires) to resolve between various competing capacities. According to this view, whenever a practice contains a constitutive rule which enables a new type of action (e.g. the practice of playing chess allowing us to perform a castling operation), if we are to perform this new action, we must have a desire which motivates us to perform it. This view is recognizably Humean.

An alternative approach is to see the practice as giving us *norms* – things we *should* do. In this case, we need our *reason* to resolve between various conflicting norms (just as it also resolves between conflicting factual claims). In this alternative view, norms explain desires. This view is recognizably Kantian.

	Hume	*Kant*
What practices give us	Capacities	Norms
How we resolve different requests	By weighing our desires	By resolving the incompatibility between norms

The problem with the Humean picture, of practices giving us the capacity to perform new actions which we use our desires to adjudicate between, is that it pushes the question back only one step further, and leaves unanswered the obvious next question – how do we decide which desires to adopt? The Humean gives no answer to this – we weigh the various actions according to the various desires, but the desires are simply *given*. In this picture, reasoning about what I should do is very different from reasoning about what is the case – reasoning about action involves *weighing* the strengths of various considerations, rather than *resolving* incompatible claims. The Humean has this depressing view of people as merely *beasts who calculate*. The alternative is that practices yield norms, and norms are evaluated according to the same canons of rationality that are used to evaluate declarative claims: we look at the *reasons* for the conflicting claims. This alternative is rationalist in that there is no aspect of decision-making which is incapable of being brought under the scrutiny of rationality.

Note that the three criticisms of the Type/Token approach have all had the same form: the Type/Token approach assumes that in order to participate in a practice, there is a capacity (scorekeeping, manifesting the

state of the practice, decision-making via the weighing of desires) which is *outside* the practice. In each case, the proposed ethnomethodological alternative is to move the capacity *inside* the practice so that it is public, witnessable and itself susceptible to normative evaluation.

The Type/Token approach is more than just the claim that it is possible to divide practices into types and tokens. It is uncontroversial that we discern certain practice-types in our culture. What is controversial and unsatisfying about the Type/Token approach is not the distinction itself, but how the distinction is used. At the heart of Type/Token approach is the idea that we understand the particular practice tokens through a prior understanding of the practice-types. It is this claim which the ethnomethodological alternative denies.

This diagram summarizes the alternative ethnomethodological approach I am recommending:
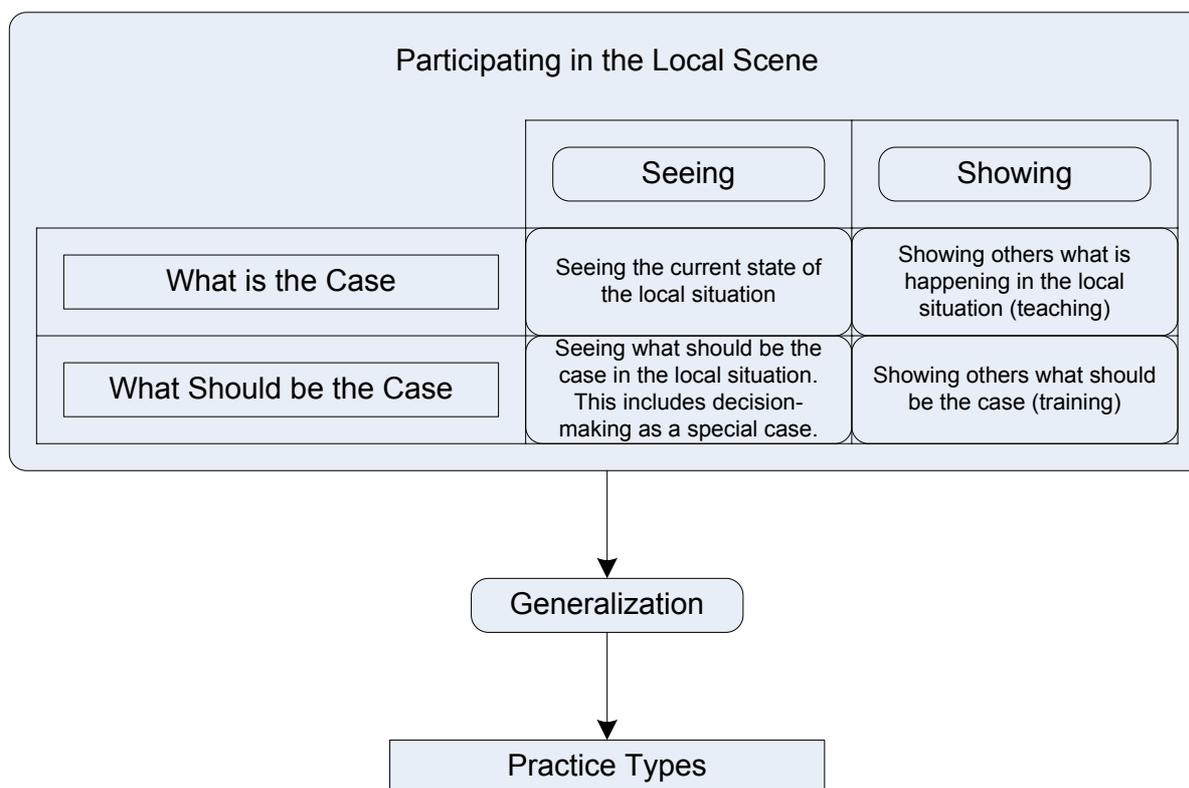
Here, scorekeeping (seeing) and manifesting the state of the scene (showing) are activities which are themselves part of that very scene. Understanding the local scene involves seeing both what is the case, and what should be the case. In this picture, practice-types are objects which we construct *subsequent* to understanding the local scene, by a process of generalization. In this picture, generalization is a process of dropping

information from the local scene. When generalizing, we must be very careful not to lose sight of the very seeings and showings which made the local scene intelligible in the first place.

7. *Recap*

I have been criticizing the Type/Token approach, and in doing so, proposing an alternative.

	Type/Token	*Alternative (Wittgenstein / Ethnomethodology)*
Scorekeeping (Seeing)	Scorekeeping is a private mental capacity.	Scorekeeping is a public act which is part of the very practice which it seeks to understand, itself subject to normative assessment.
Revelation	Revelation is the ability to divine new practice-types. It is a private mental capacity which is sharply distinct from scorekeeping.	Instead of two very different types of capacity, there is only one capacity: the ability to see the local norms in the local situation.
Manifesting the state of the practice (Showing)	Showing is a separate optional action done by teachers to novices.	Showing is a public act which is continually expected of all participants, which is itself subject to normative assessment.
Decision-making	The practice gives us the capacity to perform new actions. We evaluate these actions according to our desires, by weighing them.	The practice gives us norms. We resolve incompatible norms in the same sort of way in which we resolve incompatible declarative sentences.
Priority of practice	We understand the	We generalize to the

types to tokens	practice-token we are currently in because we understand the practice-type.	practice-type because we understand the practice-token we are in. In the act of generalization, much of the detail of the work needed to manifest the local situation is lost.
The local situation	The local situation is just the union of the practice tokens we are currently in.	The local situation is a collection of local norms.

These criticisms of the Type/Token approach have been based on an alternative proposal: the intelligibility of the local practice is an ongoing achievement, requiring constant work from the participants. This work is the manifestation of the state of the scene. It is not just the interpreter (or newcomer) who must see the state of the situation, but all participants must continually see and show the state of the scene. Their continual seeing and showing is "keeping it real", in a very real sense.

In case you were worrying that this is all getting a little too ethnomethodological, I am about to reformulate this alternative approach in a formal language.

8. *Can the Alternative Wittgenstinian/Ethnomethodological Approach Be Formalized?*

We can participate in an indefinite variety of practices. This infinite array of capacities needs to be decomposed into a finite set of capacities. I have been looking at the traditional decomposition, the Type/Token approach, and have been criticizing it on ethnomethodological and Wittgenstinian grounds. But can the ethnomethodological alternative be used as a decompositional account of the capacity to participate in practices? Are the insights of Ethnomethodology merely negative, merely showing what is wrong with the Type/Token approach, or do they point the way to an alternative decomposition of the ability to participate in practices? Is the

Wittgenstinian/ethnomethodological position merely destructive[4], or can it be used constructively?

Can we combine the ethnomethodological insights of Garfinkel/Wittgenstein with the desire for a formal decomposition?

Searle comes close to combining the ethnomethodological position with a desire for a formal decomposition. He subscribes to at least two of the central claims of the ethnomethodological alternative to the Type/Token approach:

- Showing the state of the practice is itself part of the practice. His own formulation of this is in terms of status function declarations
- Understanding a practice requires understanding the deontic norms which flow from it. It is insufficient to see the practice as yielding non-deontic capacities. To understand it, we must understand it as yielding norms.

But Searle combines these insights with the desire to decompose the capacity to participate in practices into its constituent parts: "If there is one thing we know from the cultural anthropology of the past century, it is that there is an enormous variety of different modes of social existence. The assumption I will be making, and will try to justify, is that even though there is an enormous variety, the principles that underlie the constitution of social reality are rather few in number." [*Social Ontology: Some Basic Principles*, p. 6] And again: "[Our investigation] has begun to show that the enormous complexity of the body of institutional reality has a rather simple skeletal structure. [*Social Construction of Reality*, p.112]

With the exception of Searle, ethnomethodology and formal decomposition have not, to my knowledge, been conjoined. It may be a matter of mere historical accident, but the people who advocated the ethnomethodological approach were uninterested or worse, deeply skeptical, of the formal decompositional approach.

We have been focusing on two pairs of alternatives. So there are four possibilities:

---

[4] Wittgenstein didn't seem to think that this sort of decomposition was worthwhile. (Or, perhaps, he just didn't think that such a decomposition was part of *philosophy*. Perhaps his point was merely demarcational).

	Type/Token Account of Practices	Ethnomethodology
Informal description of norms	Traditional Sociology	Garfinkel, Wittgenstein
Logical / formal description of norms	Object-Oriented Computer Models of Multi-Agent Systems	Searle (?) This proposal ☺

I am sympathetic to both the ethnomethodologists and the formalists, and – at the risk of irritating both sides – I hope to combine them.

9. *Defining a Formal Language for Describing Situations*

We are looking at ethnomethodological theories in which seeing the state of the situation and showing others the state of the situation are core parts of the very practices which they seek to describe. But unlike the authors who inspired this perspective, who are uninterested in (or skeptical of) the need to re-express this position formally, we want a formal language in which this capacity can be broken down into its constituent parts. We want a formal *language of practice*, as opposed to a language of thought.

I will now give a simple language in which this capacity is decomposed. The guiding intuition behind L is to give equal status to two pairs of complementary concepts: See/Show and Normative/Descriptive.

	Normative	*Descriptive*
See	Seeing what should be the case	Seeing what is the case
Show	Showing what should be the case	Showing what is the case

Seeing is here being used to denote a pre-linguistic form of awareness, not necessarily related to vision. Seeing is the sort of awareness that a dog has when he sees that there is a squirrel in the tree. This pre-linguistic form of Seeing is to be contrasted with fine-grained intensional states like *belief*. Scorekeeping is one particular form of Seeing — scorekeeping just is seeing the state of the local scene.

Showing is here being used to denote a pre-linguistic form of communication. This pre-linguistic form of Showing is to be contrasted with intensional speech acts like *saying*.

Given that seeing and showing are core parts of the practice, and that we are interested in cases where seeing and showing are themselves subject to normative assessment (cases where someone *should see* or *should show*), the first place to look would be some sort of combination of deontic and epistemic logic.

But such a combination would presuppose logical connectives. If it presupposed complex logical language, it would not be a foundational language. Furthermore, standard deontic logic is unsatisfactory in many ways as a formal language for modeling norms: it inherits all the problems from propositional logic to do with material implication being a weak and distorted interpretation of implication, and it introduces problems of its own. It means that, for example O(p ∨ ~p) has to be intelligible.

The language to be presented here, by contrast, is *pre-logical*: it does not include disjunction or negation or existential quantification. We cannot even express O(p ∨ q) in this language. It is intended to be the *simplest possible language* in which to describe norms of seeing and showing.

I wanted the language to be simple, and I wanted a simple name to call it. So I decided to call it L.

10. *The Syntax of L*

L is defined according to the rules:

States	$S ::= F \mid F(x_1, \ldots, x_n)$
Terms	$T ::= S \mid S.T \mid S:T$
Expressions	$E ::= T \mid See(x, J) \mid Show(x, y, J)$
Deontic Formulae	$D ::= {\uparrow}E \mid {\downarrow}E$
Conjunctions	$C ::= D \mid D \wedge C$
Judgments	$J ::= C \mid C \rightarrow C$

Here

- F is a function term, and $x_1, \ldots, x_n$ are terms referring to individual objects.
- S.T and S:T are ways of specifying sub-states of S, so we can build trees of expressions. In S.T, T is the unique child of S, in S:T, we are saying that T is one of the children of S, but there may be many others.

- ↑E says that E should be the case. ↓E says that E actually is the case.
- See is the term in L for scorekeeping the state of the practice. See(x, J) means that agent x sees that J. J itself may be either normative (↑) or descriptive (↓).
- Show is the term in L for manifesting the state of the practice. Show(x, y, J) means that agent x shows agent y that J. Again, J itself may be either normative or descriptive.

The richness of L derives from the recursive clauses for Terms, Expressions, Conjunctions and Judgments.

Here are some typical terms in L:

> X
> X.Y
> Move(x)
> Game(x, y).Move(x)
> Game(x, y):Score(x).2
> Game(x, y):Score(y).1

In X.Y, we say X is the *parent* of X.Y.

The complete expressions in L are the members of J – the complete judgments. Here are some examples of expressions in J:

↑In(a, b)
↓In(a, b)
> ↓Game(x, y).WhoseMove(x)
> ↓Game(x, y).Move(x) → ↑Play(x)
> ↓Game(x, y).Move(x) ∧ ↓Play(y) → ↓Game(x, y).Fault(y)
> ↓Game(x,y).Fault(y) → ↑Rebuke(x, y)

↓See(x, ↑In(a, b))
↓See(y, ↓In(a, b))
↓In(a, b) → ↑See(x, ↓In(a, b))
↓See(x, ↑See(y, ↓In(a, b)))

Here are some expressions which are *not* in J:

In(a, b)
↑↑In(a, b)
↓~F
~↑F
↓F ∨ ↓G

In L, there are four kinds of arrows between the two realms:

$$\downarrow X \rightarrow \downarrow Y$$
$$\downarrow X \rightarrow \uparrow Y$$
$$\uparrow X \rightarrow \uparrow Y$$
$$\uparrow X \rightarrow \downarrow Y$$

L sees the world as divided into two planes, Up and Down, with arrows within and between the planes:
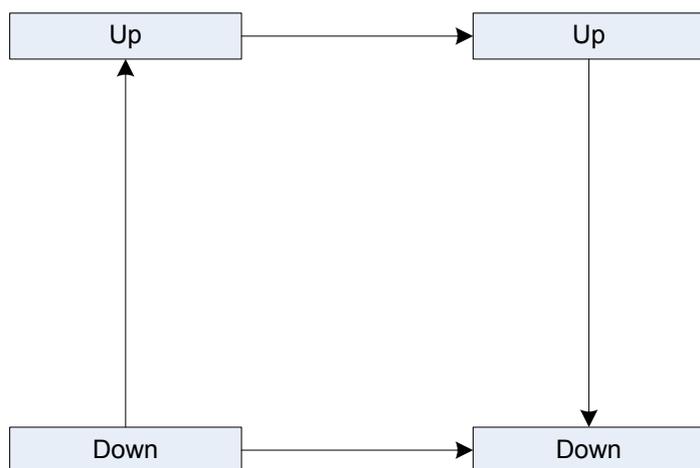
11. *Examples of Expressions in L*

We need a nice simple case to illustrate how L works. Here is one of Harvey Sacks' favorite examples: "The baby cried. The mommy picked it up":

$\downarrow$Cry(baby)	*The baby is crying*
$\downarrow$See(mommy, $\downarrow$Cry(baby))	*The mommy sees that the baby is crying*
$\downarrow$See(mommy, $\downarrow$Cry(baby) $\rightarrow$ $\uparrow$Pickup(mommy, baby))	*The mommy sees that she should pick up the baby if it is crying*
$\downarrow$See(mommy, $\uparrow$Pickup(mommy, baby))	*The mommy sees that she should pick up the baby*
$\downarrow$Pickup(mommy, baby)	*The mommy picks up the baby*

The crucial move in this case is when the mommy realizes that the baby's crying means that she should pickup the baby:

$\downarrow$See(mommy, $\downarrow$Cry(baby) $\to$ $\uparrow$Pickup(mommy, baby))

This is the mommy deriving an ought ($\uparrow$) from an is ($\downarrow$). Arrows between $\downarrow$ and $\uparrow$ are endemic in L.

This first example provides no explanation for why the baby started crying. Was his crying an automatic response to wind, or was it a communicative act, expressing his need for attention and his desire to be picked up? At some point, toddlers move from the first to the second, and when they do, it is a profound conceptual shift. For example:

$\downarrow$See(toddler, $\downarrow$Lonely(toddler))	*The toddler sees that he is lonely*
$\downarrow$See(toddler, $\downarrow$Lonely(toddler)$\to$ $\uparrow$Pickup(mommy, toddler))	*The toddler thinks that he should be picked up when he is lonely*
$\downarrow$See(toddler, $\uparrow$Pickup(mommy, toddler) $\to$ $\uparrow$See(mommy, $\uparrow$Pickup(mommy, toddler)))	*The toddler sees that mommy should realize that she should pick up the toddler if the toddler should be picked up*
$\downarrow$See(toddler, $\uparrow$See(mommy, $\uparrow$Pickup(mommy, toddler)))	*The toddler sees that the mommy should realize that she should pick up the toddler*
$\downarrow$See(toddler, $\uparrow$See(mommy, $\uparrow$Pickup(mommy, toddler)) $\to$ $\uparrow$Show(toddler, mommy, $\uparrow$Pickup(mommy, toddler)))	*The toddler sees that he should show mommy that he should be picked up if he wants her to see that he should be picked up*

↓See(toddler, ↑Show(toddler, mommy, ↑Pickup(mommy, toddler)))	*The toddler realizes he has to show mommy that he should be picked up*
↓See(toddler, ↑Show(toddler, mommy, ↑Pickup(mommy, toddler)) → ↑Cry(toddler))	*The toddler realizes that he should cry if he needs to show mommy that he should be picked up*
↓See(toddler, ↑Cry(toddler))	*The toddler realizes he needs to cry*

One crucial move in this case is the toddler's reasoning based on his understanding of what his mommy *should see*:

↓See(toddler, ↑See(mommy, ↑Pickup(mommy, toddler)))

The other critical move is when the toddler reasons from what his mommy should see to what he should show:

↓See(toddler, ↑Show(toddler, mommy, ↑Pickup(mommy, toddler)))

L was designed to express what people do and should do when they participate in such situations – how they see what others see and should see (in a pre-linguistic non-intensional sense of see) and how they communicate with each other (in a pre-linguistic sense of communication).

11. *How L is Different from Standard Deontic Logic*

L is very different from Standard Deontic Logic (SDL). SDL is a modal logic which extends propositional logic with a non-truth-functional operator O (Ought), which can apply to any sentence of SDL.

The O operator has the usual axioms of a modal logic

$$O(p{\rightarrow}q) \rightarrow (O(p) \rightarrow O(q))$$
$$\sim(O(p) \wedge O(\sim p))$$

It has the usual inference rules of a modal logic

Modus Ponens
Necessitation: if we can infer p, then we can infer O(p)

Standard Deontic Logic isn't really a logic of its own — it's just a *reinterpretation* of a standard modal logic K+D, reinterpreting the box operator as Ought.

There are four main differences between SDL and L. Firstly; L has no negation or disjunction. L tolerates no vagueness or indecision. Unlike SDL, it is a language of specific determinate claims. Secondly, SDL has the implication arrow $\Rightarrow$ of material implication whereas L uses the $\rightarrow$ arrow of *defeasible* implication. Thirdly, SDL allows arbitrary iteration of O and P in sentences like O(O(P(q))), whereas in L, iteration of $\uparrow/\downarrow$ is *alternated* with See/Show. Finally, SDL assumes an antecedently-intelligible core of declarative propositional sentences, whereas in L, $\uparrow$ and $\downarrow$ are coeval. I will elaborate on each of these.

12. *L is a determinate language containing no negation or disjunction*

L, unlike SDL, is a *determinate* language. If we want to say that x is not red in L, we have to say something specific about the color of x. Now the negation of p is the weakest claim which is incompatible with p, but L is interested in stronger claims.

The disjunction of p and q is the strongest claim which is implied by both p and q. Disjunction is a general operation, which, applied to any two expressions, will yield the strongest claim which is vaguer than both. Like negation, disjunction is a vagueness-creating operator. In L there is no general operation on expressions which makes them vaguer. There is local vagueness: we can make a particular norm tree with local vagueness. A.B and A.C contain A, and A can imply Z. But this is much more determinate than B $\vee$ C $\rightarrow$ Z

The disjunction of p and q is the strongest claim which is entailed by both p and by q:
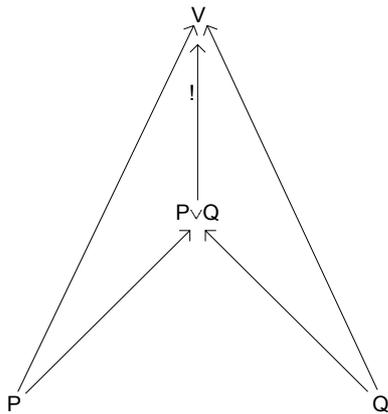
Recall that the *parent* of A.B is defined as A. Just as disjunction is the categorical sum in propositional logic, so the parent of an expression is the categorical sum in L. A.B is the strongest claim which is implied by both A.B.P and by A.B.Q:
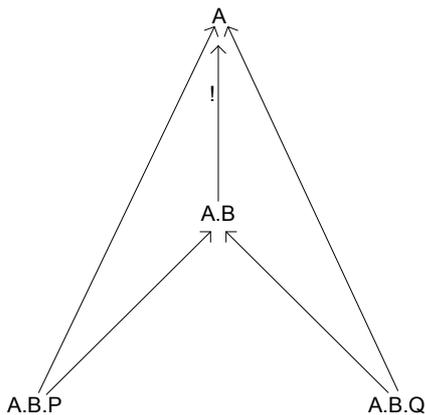
12. *Defeasible implication*

The → arrow in L is defeasible implication, not material implication.

Because material implication $p \Rightarrow q$ is defined as equivalent to disjunction ($\sim p \lor q$), and because we can always add extra disjuncts to a disjunction while preserving truth, we can move from ($\sim p \lor q$) to ($\sim p \lor \sim r \lor q$) which is equivalent to $p \lor r \Rightarrow q$. Strengthening the input is always valid with material implication.

Not so with defeasible implication. We cannot infer from

$\downarrow$Bird(x) $\rightarrow$ $\downarrow$CanFly(x)

to:

$\downarrow$Bird(x) $\wedge$ $\downarrow$Penguin(x) $\rightarrow$ $\downarrow$CanFly(x)

P $\rightarrow$ Q should be interpreted as: P means that Q, or: P provides a reason for Q. Adding arbitrary additional conjuncts does not preserve reason-givingness.

13. *Iteration of $\uparrow$/$\downarrow$ is alternated with See/Show*

Standard deontic logic allows iteration of O and P. E.g.
O(O(O(q))) and P(P(P(q))) and O(P(O(q))) etc.

It is unclear, at first glance, what sense to attach to O(O(q)). Like O(p $\vee$ q), it seems more of a product of the syntactic machinery of SDL than something which is antecedently intelligible.

Likewise, SDL allows O outside $\rightarrow$, so we can say O(p $\rightarrow$q), as distinct from O(p) $\rightarrow$ O(q). L does not allow this: arrows in L cannot be prefixed with $\uparrow$ or $\downarrow$.

14. *Up($\uparrow$) and Down($\downarrow$) are coeval, equiprimordial*

Deontic logics are formed by adding extra non truth-functional operators to a propositional logic. They are based on the assumption that there is a self-contained practice of saying how the world is, and then – as an optional extra – we can say how the world should be. But I believe this deeply-ingrained assumption is false — there is no autonomous discursive practice[5] in which all we can do is say how things are. This highly controversial claim needs justification.

Inferentialists are fond of pointing out that we don't understand an expression unless we know various inferences involving that term. Typically, we imagine inferences in the descriptive realm: inferences from what is the case to what is the case. But a stronger inferentialist claim is that we don't understand an expression unless we can also inferentially connect it to what *should* be the case. This stronger claim is *normative inferentialism*: "You do not understand a claim unless you know (defeasible) inferences which

---

[5] The term is Brandom's: an *autonomous discursive practice* is a language-game you could play, though you played no other

(transitively) connect that claim to both the realms of what is the case (↓) and also what should be the case (↑)."

I shall consider a number of examples which individually suggest normative inferentialism, to get the gentle reader in the mood, and then I will provide an argument for why *all* terms must be (indirectly) inferentially connected to Up – the realm of what should be the case.

Our first example to support normative inferentialism is a standard functional term: "You don't understand what a car is unless you know that: x is a car implies x *should* move forward when you press the right buttons."

Another example, of a functional term applied to a person: "You don't understand what a firefighter is unless you know that: x is a firefighter implies x *should* put out the fire when there is a fire nearby."

The classic example of normative inferentialism is belief. Ascribing belief is implicitly normative: "You don't understand what a belief that [y is red] is unless you know that: agent x believes that [y is red] implies x *should* believe that [y is colored]."

What is true for material inferences is similarly true for material incompatibility relations: "You don't understand what a belief that [y is red] is unless you know that: agent x believes that [y is red] implies x *should* not believe that [y is green]."

Here is Brandom making the point about intentional states: "The starting point of his [Wittgenstein's] investigations is the insight that our ordinary understanding of states and acts of meaning, understanding, intending, or believing something is an understanding of them as states and acts that *commit* or *oblige* us to act and think in various ways. The meaning of an linguistic expression must determine how it would be *correct* to use it in various contexts. [The content of a particular belief] determines how it is *appropriate* for it to be related. A particular intention may or may not settle how one *will* act, but its content determines how it is *appropriate* to act. [*Making It Explicit*, p. 13]"

In response to cases like these, philosophers typically nod wisely, and say that these expressions are normative. But what is so special about these expressions? How do we know which subset of our language has this special normative import? Normative inferentialism is the generalization of this claim to *all* expressions.

The strong and controversial claim which underpins L is that ↑ and ↓ are coeval – there is no autonomous discursive practice which involves only ↓. I have given examples of particular expressions which we don't understand

unless we can inferentially connect them to ↑. But why should it hold for all expressions?

The justification for normative inferentialism is that a collection of ↓ judgments, taken on their own, are completely inert. You can do nothing with them. We need ↑ judgments to connect to action! Recall the distinction between theories of practice which use capacities and theories which use norms. If we see practices as giving us new capacities, then we can believe in an independently intelligible realm of ↓, because it is our *desires* which can determine which capacities we follow. But if we see practices as giving us norms, we need to have at least one norm to act. So a term which had no inferential connection to the norms would have no effect on what we do – it would be inert: "A wheel that can be turned though nothing else moves with it, is not part of the mechanism." [*Philosophical Investigations*, §271]

Someone who used a language which just involved ↓ expressions would be, in Dummett's memorable phrase, an "intelligent and sentient tree", who could observe the world and utter sounds, but could engage in no other type of action. But the situation is even worse than Dummett imagines, for the sentient tree could do nothing at all. Observing the world and uttering sounds are *actions*. You cannot perform an action unless you have a *reason* for action, and the sentient tree, which has no connection from ↓ to ↑, has none.

What about an everyday term like "red"? What are the inferential connections between "red" and ↑? Here are a couple of inferential connections to ↑ which are very culturally-specific:

$$\downarrow Red(x) \wedge \downarrow TrafficLightControls(x, y) \rightarrow \uparrow Stop(y)$$

$$\downarrow Bus(x) \wedge \downarrow London(x) \rightarrow \uparrow Red(x)$$

Now clearly it is possible to understand a term like "red" without knowing these particular inferential connections to Up, but – nevertheless – the claim is that you must know *some* inferential connections to Up, even if they are not these particular ones.

The world is unavoidably a mixture of ↑ and ↓. We cannot get below to the solid ground of just ↓. This is what Wittgenstein meant when he wrote:
"The difficult thing here is not, to dig down to the ground; no, it is to recognize the ground that lies before us as the ground. For the ground keeps on giving us the illusory image of a greater depth, and when we seek to reach this, we keep on finding ourselves on the old level. Our disease is one of wanting to explain." [*Remarks on the Foundations of Mathematics,* §30]

Wittgenstein is saying that the bedrock explanation is the world of two realms, descriptive and normative, and there is no deeper level, just in terms of the descriptive, which is self-sufficiently intelligible.

The normative and the descriptive are, to use a well-worn expression, two sides of one coin: neither intelligible without the other.

Wittgenstein's well-known example of the builder[6] might be interpreted as an example of an autonomous discursive practice involving ↑ but not ↓. In this example, the builder asks his assistance to bring objects of various types. The builder just says "Slab" (and in this case, he means ↑Slab), and the assistant brings the right type of object. Wittgenstein says "Conceive of this as a complete language game". But suppose the assistant brings something that is not a slab. Then the builder will want to *correct* the helper. This correction will involve the builder getting the assistant to see that the thing he brought was not a slab. This involves ↓. Resolving failures of communication requires using both ↑ and ↓.

15. *How L Handles Some of the problems of Standard Deontic Logic*

15.1. *Obligations of Disjunctions*

Because SDL extends propositional logic, and allows O to apply to any arbitrary propositional formula, it allows expressions like $O(p \vee q)$ and $O(p \rightarrow q)$. Because it inherits its semantics from K+DL, the following are always valid:

$$O(p \vee \sim p)$$
$$O(p \rightarrow p)$$

We are obligated to make it the case that tautologies hold! This seems a heavy burden indeed.

Further, SDL allows us to infer from $O(p)$ to $O(p \vee q)$, for any arbitrary q — no matter how unsavory!

The problem here is at the root, with the very sentences that SDL allows as syntactically acceptable. Just because a formal language accepts a certain sentence, it doesn't mean that we can do anything with it, that we can make it mean anything. It is not clear what sense we could make of $O(p \vee q)$, $O(p \rightarrow q)$ or $O(O(p))$. These are not formalized versions of expressions which are

---

[6] *Philosophical Investigations*, §2

antecedently intelligible. It is rather that they are forced on us by the syntactic machinery of SDL, and we don't know what to do with them.

"Philosophers are often like little children who scribble some marks on a piece of paper and then ask the grown-up." ["What does this mean?" *Culture and Value*, p. 17]

Von Wright, the founder of deontic logic, originally had a formalism which was much more syntactically restricted. In this earlier version, the deontic operator (he started using P as primitive) was applied to *actions*, not to sentences. This meant that it was impossible to express $O(p \vee q)$, $O(p \rightarrow q)$ or $O(O(p))$. Under pressure from his colleagues, he moved to the formalism we now know as SDL.

The contested expressions $O(p \vee q)$, $O(p \rightarrow q)$ and $O(O(p))$ are not allowed in L. In L, there is no disjunction whatsoever, so we cannot even express $O(p \vee q)$ at all. Further, in L, expressions of the form $p \rightarrow q$ cannot be embedded inside ↑, so we cannot say $\uparrow(p \rightarrow q)$. This syntactic minimalism is a virtue. In this respect, L is *intermediate* between Von Wright's original logic (where deontic operators were applied to actions) and his later logic (where deontic operators were applied to arbitrarily complex logical propositions).

15.2. *Chisholm's puzzle*

Standard Deontic Logic has a number of problems when it tries to handle cases involving contrary-to-duty conditionals (cases which include an if-then clause in which the antecedent involves violating a norm).

Consider the following example:

1. It ought to be that a certain man goes to the assistance of his neighbors
2. It ought to be that if he does go, then he ought to tell them he is coming
3. If he does not go, then he ought not tell them he is coming
4. He does not go

The intuitive conclusion is:

5. He ought not tell them he is coming

These sentences all seem to be logically independent: no one of them follows from the others.

The trouble is there is no plausible translation of these sentences into standard deontic logic, which yields the intended conclusion and which preserves their logical independence.

The obvious translation is:

O(A)
O(A → T)
~A → O(~T)
~A

(Where A is the man assists the neighbors, and T is the man tells them he is coming).

In this translation, we can derive both O(T) and O(~T) – a contradiction.

The only other plausible translation doesn't fare much better:

O(A)
A → O(T)
~A → O(~T)
~A

This no longer yields a contradiction, but it loses the independence of the four propositions. Now the second is a direct consequence of the fourth (because the arrow of Standard Deontic Logic is the arrow of material implication, where p → q is equivalent to ~p ∨ q, which follows directly from ~p).

We have no such problem in L, because the arrow in L is the arrow of defeasible implication, not the arrow of material implication. In L, this situation is modeled as:

↑ A
↓ A → ↑T
↓~A → ↑~T
↓~A

(Note we are using local negation ~ as syntactic sugar).

In this formulation, ↑~T follows as we should expect. Further, the four sentences are independent: ↓A → ↑T does not follow from ↓~A.

16. *Two senses of permission*

Two distinct notions of permission have been distinguished in the deontic logic literature. The trouble is that standard deontic logic can only express one of them.

If agent x is *weakly*-permitted to do A, then there is no reason why he shouldn't do A. Weak permission is inter-definable with obligation:

O(x, A) iff ~P(x, ~A)

To say that agent x is *strongly*-permitted to do A, by contrast, is to say more than just that he is weakly permitted. It is to say that someone or something actually permitted him to do it – an action was performed that *showed* everyone that it was ok for him to do it. Strong permission effectively forbids any future legislation which rules out the doing of A. Weak permission looks to the past, whereas strong permission is itself normative over the future.

Strong permission cannot be expressed in Standard Deontic Logic – but we can express both concepts naturally in L:

↓WeaklyPermitted(x, A) iff ↓~See(x, ↑~Perform(x, A))

↓StronglyPermitted(x, A) iff ↑~See(y, ↑~Perform(x, A)), for all y

If I am weakly permitted to do A, then I can't see any reason why I shouldn't perform A. But if I am strongly permitted to do A, then nobody should think I shouldn't do A.

17. *Reasoning with Conditional Norms*

Input/Output logic was introduced to handle the sorts of problems we have been outlining with Standard Deontic Logic. Input/Output logic uses a different notation from L. It does not have any deontic operators like O or ↑ — it just has propositions and conditions. An inference from a to b is represented as (a, b).

Input/Output logic avoids most of the puzzles associated with standard deontic logic, but it introduces different problems of its own. Most notoriously, if we have two conditions (a,x) and (x,y), should we be able to

infer (a,y)? There are certain examples where the inference is clearly invalid. Suppose we have:

a	I owe my neighbor $10
x	I pay him back $10
y	My neighbor thanks me

If we let the inference go through, we will conclude that if I owe my neighbor $10, he should thank me. But shouldn't he only thank me if I have *in fact* paid him, not just because I *should* pay him?

L diagnoses the problem here as a reflection of the impoverishment of Input/Output logic's expressive resources. Input/Output logic cannot distinguish between something being the case, and something that should be the case. But in L we can distinguish between

$$\downarrow A \rightarrow \uparrow X$$
$$\uparrow X \rightarrow \uparrow Y$$

and:

$$\downarrow A \rightarrow \uparrow X$$
$$\downarrow X \rightarrow \uparrow Y$$

In the first pattern, the inference is perfectly valid. In the second pattern, the inference is invalid. The example of the $10 debt falls squarely under the second pattern.

18. *Using L to Express the Situations We Find Ourselves In*

So far we have introduced and motivated a formal language, L, for describing what people do and should do. We have done the work of introducing and motivating the language, and now it is pay-back time — What do we gain by using L? What can we say in it?

18.1. *Expressing Directions of Fit in L*

In *Foundations of Illocutionary Logic*, Searle writes: "In spite of frequent philosophical protestations[7] to the contrary, there is a rather limited number of things one can do with language."

He uses the notion of direction-of-fit: "Intuitively the idea of the direction of fit of an utterance can be clarified by pointing out that if the propositional content fails to match reality, one side or the other is at fault. If my statement fails to match reality, it is my statement and not reality that is at fault. Statements can be said to be true or false, and statements and other members of the assertive class are said to have the word-to-world direction of fit. But if my order is disobeyed or my promise is not carried out, it is not my order or promise which is at fault but rather reality in the person of the hearer who disobeyed the order or myself who failed to carry out the promise. Such utterances are said to have the world-to-word direction of fit."

He says there are exactly four directions of fit:

> 1. The word-to-world direction of fit.
> In achieving success of fit, the propositional content of the illocution fits an existing state of affairs in the world.
> 2. The world-to-word direction of fit.
> In achieving success of fit, the world is altered to fit the propositional content of the illocution.
> 3. The double direction of fit.
> In achieving success of fit the world is altered to fit the propositional content by representing the world as being so altered.
> 4. The null or empty direction of fit.
> There is no question of achieving success of fit between the propositional content and the world, because in general success of fit is presupposed by the utterance.

---

[7] See Wittgenstein, *Philosophical Investigations*, §23

The double-direction of fit is prima-facie puzzling. For the two straightforward directions of fit make sense because we know what should be altered when the words and world are out of sync. In a word-to-world direction of fit, the words should be altered. In a world-to-word direction of fit, the world should be altered. But in a double-direction of fit, what should be altered now?

When we express these directions of fit in L, this initial puzzlement subsides.

All speech-acts are done for the reason of showing something. All speech-acts satisfy the expression:

$$\uparrow\text{Show}(x, y, \_) \rightarrow \uparrow\text{Do}(x, P)$$

In other words, one reason for x performing the speech act P is to show y something.

What sort of expressions can fill the blank in this formula? The different ways of filling the blank will correspond to the different directions of fit of utterances. For example:

Direction	*Intention-in-action*
Word-to-World	$\uparrow\text{Show}(x, y, \downarrow N) \rightarrow \uparrow\text{Do}(x, P)$
World-to-Word	$\uparrow\text{Show}(x, y, \uparrow N) \rightarrow \uparrow\text{Do}(x, P)$
Double	$\uparrow\text{Show}(x, y, \downarrow\text{Do}(x, P) \rightarrow \uparrow N) \rightarrow \uparrow\text{Do}(x, P)$
Null	$\uparrow\text{Show}(x, y, \downarrow N \rightarrow \uparrow\text{Do}(x, P)) \rightarrow \uparrow\text{Do}(x, P)$

In the word-to-world direction of fit, agent x should do action P because he needs to show y that N is the case. For example, the agent's P action is his saying to y that "there is a storm coming", and $\downarrow N$ is the fact that the storm is coming.

In the world-to-word case, agent x should do P because he needs to show agent y that N should be the case. For example, the agent's P action is his saying "slab" to y, and $\uparrow N$ is that y should bring a slab.

18.2. *Expressing the Double Direction of Fit in L*

In the case of the double-direction of fit, L allows us to see clearly the two different arrows involved: agent x should perform P because he needs to show

y that, *by the very performance of P,* he is making it the case that N should be the case:

$$\uparrow Show(x, y, \downarrow Do(x, P) \rightarrow \uparrow N) \rightarrow \uparrow Do(x, P)$$

Searle considers status-assignments of the form:

X counts as Y in C

For example: "Saying "I appoint you chairman" means that you are the chairman if spoken by someone with the appropriate authority in the context of an appointment-ceremony. A normative consequence of this status of being a chairman is: if you are chairman, then you are responsible for opening the next meeting." [*The Construction of Social Reality*, p.54]

In L, this would be rendered:

$$\downarrow Do(x, X) \wedge \downarrow C \rightarrow \downarrow Y(y)$$

In our example, the X action which makes it the case that the Y status holds is itself a speech-act. In such cases, the X action is a performative and has the double-direction of fit described above:

$$\uparrow Show(x, y, \downarrow Say(x, y, \text{"I appoint you chairman"})) \rightarrow \downarrow Chairman(y))$$
$$\rightarrow \uparrow Say(x, y, \text{"I appoint you chairman"})$$

In other words: the reason why I say "I appoint you chairman" is to show you that, by the very act of saying those words, I have appointed you chairman.

These new statuses have normative consequences. In our example:

$$\downarrow Chairman\ (y) \rightarrow \uparrow OpenMeeting(y)$$

This is the general logical form of status-function declarations:

$$\uparrow Show(x, y, \downarrow Do(x, P) \rightarrow \downarrow N) \rightarrow \uparrow Do(x, P)$$
$$\uparrow Show(x, y, \downarrow Do(x, P) \rightarrow \uparrow N) \rightarrow \uparrow Do(x, P)$$

We have two types, depending on whether N is in Up ($\uparrow$) or Down ($\downarrow$).

18.3. *Expressing the "Null" Direction of Fit in L*

In the case of the null[8] direction of fit, Searle says that the action *presupposes* the truth of a claim, but does not assert it. For example: thanking somebody after a marvelous dinner is an act which presupposes the dinner, but does not assert its existence (or its marvelousness). This sort of example is rendered in L as:

$$\uparrow\text{Show}(x, y, \downarrow N \to \uparrow\text{Do}(x, P)) \to \uparrow\text{Do}(x, P)$$

One of our reasons for doing P is to show that our doing P was prompted by N being the case. For example: I thanked my host for the meal in order to show her that the marvelous meal prompted my thanking. Or: I apologized in order to show that my being late meant that I needed to apologize[9]:

$$\uparrow\text{Show}(x, y, \downarrow\text{Late}(x) \to \uparrow\text{Apologize}(x, y)) \to \uparrow\text{Apologize}(x, y)$$

18.4. *Expressing Iteration in L*

Searle stresses that the structure "X counts as Y in C" can be iterated: "In such cases the X term at a higher level can be a Y term from an earlier level. For example, only a citizen of the United States as X can become a President as Y, but to be a citizen is to have a Y status function from an earlier level. It is no exaggeration to say that these iterations provide the logical structure of complex societies. [*The Construction of Social Reality*, p.80]

This sort of iteration can be described simply in L:

$$\downarrow\text{BornLegallyIn}(x, c) \to \downarrow\text{Citizen}(x, c)$$
$$\downarrow\text{Citizen}(x, c) \wedge \downarrow\text{WinsElection}(x, c) \to \downarrow\text{President}(x, c)$$

---

[8] Searle has recently revised his terminology – instead of calling it the "null" direction of fit, he now calls it the *presuppositional* direction of fit. For the reasons given in the text below, this terminological change is a nice improvement.

[9] This analysis is not meant as an *analysis* of apologizing. It does not, after all, capture the sincerity condition at the heart of the apology – that the speaker is expressing his regret. Instead of giving an analysis, we are locating the apology within the space of norms by giving a reason for the action: *one* of the reasons for making the apology was to show that the situation warranted the apology.

Each of these Y terms has deontic responsibilities:

$\downarrow$Citizen(x, c) $\rightarrow$ $\uparrow$PayTaxes(x, c)
$\downarrow$President(x, c) $\rightarrow$ $\uparrow$ProtectInterestsOf(x, c)

This sort of iteration is *arrow iteration*, which involves inferences using the transitivity of $\rightarrow$.

But there is also *another* type of iteration available in L. This is iteration within the epistemic operators See/Show and the arrows $\uparrow$/$\downarrow$. When performing a linguistic act, we are always showing *something*. So let us look at all possible expressions of the form

$\uparrow$Show(x, y, _) $\rightarrow$ $\uparrow$Do(x, P)

Because L has recursive structure, there is an *indefinite* number of expressions which can be placed in the blank. Here is a small fragment of the expansion tree:
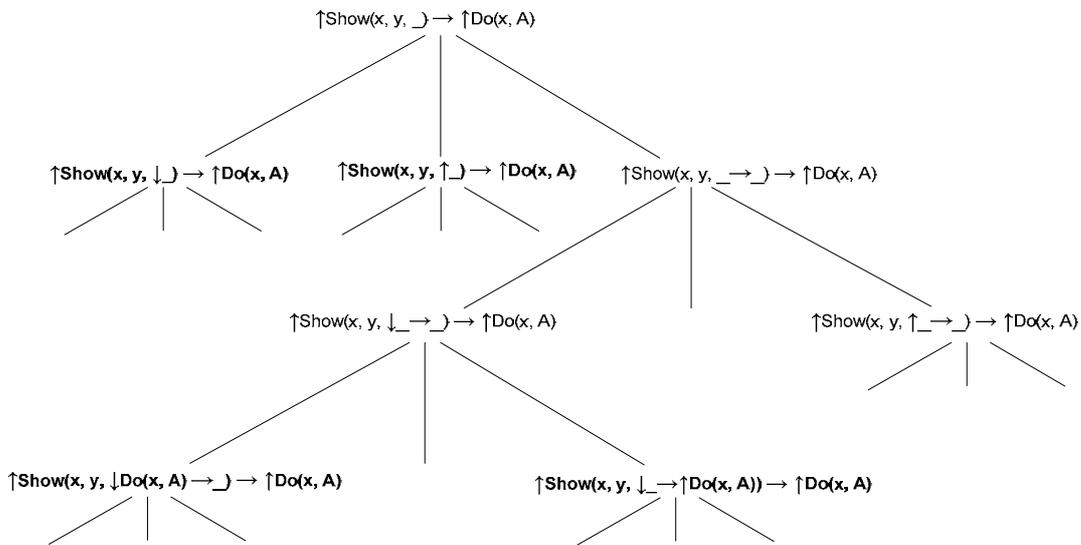
Note that Searle's four directions of fit occur in this expansion tree, represented in bold. But these are four expressions amongst indefinitely many others.

19. *Describing the Game of Giving and Asking for Reasons in L*

In *Making It Explicit*, Robert Brandom produced a remarkably specific description of the structure of the social practices needed to institute language-understanding. Following Sellars, he called this practice the Game of Giving and Asking for Reasons (GOGAR). Like Sellars, he uses the core notions of material inference and material incompatibility, but he goes further in distinguishing two fundamental deontic statuses: Commitment and Entitlement. Commitment tracks the sentences which a speaker is committed to, and Entitlement tracks the sentences which the agent doesn't need to justify. In chapter 3 of <u>Making It Explicit</u>, he describes some clear rules for how to update Commitment and Entitlement as speakers make claims.

We can re-describe GOGAR in L. Here are the terms needed:

Assert (x, p)	Agent x has asserted sentence p
CommittedTo(x, p)	Agent x has committed to p
EntitledTo(x, p)	Agent x is entitled to p
CommitmentImplies (p, q)	There is a commitment-preserving material inference from sentence p to q
EntitlementImplies(p, q)	There is an entitlement-preserving material inference from p to q
Incompatible(p, q)	Sentence p and q are materially incompatible
Justify(x, p, q)	Agent x has justified p with q
Retract(x, p)	Agent x has retracted his assertion that p

Here are the update rules of GOGAR as described in *Making It Explicit*, chapter 3:

↓Assert(x, p) → ↓Committed(x, p)   *If agent x asserts sentence p, then he is committed to p*
↓Committed(x, p) ∧ ↓Commitment Implies(p, q) → ↓Committed(x, p)   *If agent x asserts sentence p, and p materially implies q, then he is also committed to q*
↓Committed(x, p) → ↓Entitled(x, p)   *By default, agents are entitled to their commitments*
↓Committed(x, p) ∧ ↓Committed(y, q) ∧ ↓Incompatible (p, q) ∧ ↓Entitled(x, p) → ↓~Entitled(y, q)   *If two claims are incompatible, the two agents asserting them cannot be entitled to both[10]*
↓Committed(x, p) → ↑Entitled(x, p)   *If an agent is committed to a claim, then he should be entitled to it*
↑Entitled(x, p) ∧ ↓CommitmentImplies(q, p) ∧ ↓Entitled(x, q) → ↑Justify(x, q, p)   *If an agent needs to justify p, and he is already entitled to a claim q which commitment-implies p, then he should justify p with q*
↓Committed(x, p) ∧ ↓~Entitled(x, p) → ↑Retract(x, p)   *If an agent is committed to a claim he is not entitled to, he should retract it*

Note in the interests of space, we have simplified the discussion to include only inferences from one statement to another (when in fact the input to an inference is a *set* of statements), and incompatibility between pairs of sentences (when in fact incompatibility can be between sets of more than two sentences).

---

[10] We have used a slightly difference conception of entitlement from Brandom: he analyzes incompatibility between p and q as: Commitment to p precludes Entitlement to q.

20. *Expressing Phenomenological Detail in L*

Garfinkel uses a number of practical examples to show his students how traditional sociological method passes over the phenomenological detail of the local situation. One particularly striking example is the summoning phones. He is emphatic that you can only truly understand this example by participating in it, not just by reading about it. Nevertheless I will re-describe it in text, hoping powerlessly that the gentle reader will actually try it for himself.

He asks his students to tape-record five different types of phone-call:

- A phone summoning you
- A phone summoning another
- A phone *simulating* summoning you (a case where you have asked someone to ring you on your phone at a specified time, for the purpose of tape-recording the noise, but you have no intention of answering)
- A phone simulating summoning another
- A phone which is just ringing, summoning nobody

Each student is asked to tape record five examples of each of the five types of phone-call, and bring them to the next class. Then, when the class begins, Garfinkel produces his own tape and plays some examples: here is an example of a phone summoning me; later, here is one of a phone simulating summoning another.

Of course, when played back, all the tape-recorded phone-calls sound the same. The phenomenological detail has been lost. Garfinkel specifies exactly what has been lost.

When the phone is summoning me, the first silence (the silence before the first ring is heard) isn't heard until the first ring is heard. The first ring is heard coming out of a silence that is only now hearable as preceding it. When it is heard, the first ring is directed to a fixed place ahead: the moment of the second ring. Further: when the first ring is heard, it is hearably summoning me: I see that I should answer it. That I should answer it is observable to me and to others.

When the phone is summoning another, what makes it sound different, to me and to others, is that it is observable that another should answer it. The phenomenological difference corresponds exactly to a difference in the normative status.

When the phone is simulating summoning me (when I have prearranged with someone else that they will phone me at this particular time), it feels very different: this time I am aware of the initial silence (the silence before the first ring) from the very start, because I know that my friend is going to ring me, and I am anticipating that first ring. When the first ring comes, I do not hear it summoning me – I do not see that I should answer it – it is just ringing. The phone simulating summoning another is similar.

When the phone is ringing, but not ringing for anybody in particular, the phenomenological details are different again: the initial silence is not anticipated – we did not expect it to ring. When it rings for the first time, this ring has no normative import: nobody should answer it.

A phone summoning you sounds very different from one which is summoning another. This phenomenological difference can be captured very precisely in L: it is the difference between seeing that ↑Pickup(phone, me) and ↑Pickup(phone, other). A phone summoning you sounds very different from a phone *simulating* sounding you; this phenomenological difference, too, can be captured exactly in L: it is the difference between ↑Pickup(phone, me) and ↑See(me, ↑Ring(phone)).

The phenomenological differences are summarized in the table:

	Silence	First Ring
Summoning me		↑Pickup(phone, me)
Summoning another		↑Pickup(phone, other).
Simulating summoning me	↑See(me, ↑Ring(phone))	
Simulating summoning another	↑See(other, ↑Ring(phone))	
Phone just ringing		

Let's abandon caution for a moment. Generalizing wildly and irresponsibly from this simple example – what if all phenomenological differences were differences in how the normative was perceived?

What if feeling a desire to X based on a ground P just is perceiving that one should X if P? For example, the felt urgency of hunger is captured by:

↓See(↓EmptyStomach(me) → ↑Eat(me, carrot))

In this approach, desires are explained by norms, not norms by desires. In this approach, there is a normative judgment behind every desire, but there

are also normative judgments that do not correspond to desires at all. As Searle puts the point: "What we have in society is a set of deontic power relations. But again, one might ask the question, why should we care about these deontic power relations? Who gives a damn about my rights, duties and obligations? The answer is important: What we are discussing here are reasons for action, and to recognize something as a right, duty, obligation, requirement, etc is to recognize a reason for action. Furthermore it is a specific kind of reason for action that is absolutely essential to human society and which, as far as I can tell, does not exist in the animal kingdom: *These deontic structures make possible desire-independent reasons for action.* [*Social Ontology: Some Basic Principles*, p.10]

In *Theory of Communicative Action*, Habermas distinguishes four types of action: teleological (instrumental, desire-based), normative, dramaturgical (expressing one's internal self-image), and communicative. What are the explanatory relations between these four types of action? Are they equiprimordial, or can one explain the others? According to the Humeans, the teleological explains the others: there is no action done according to a norm unless there was a desire to follow that norm; there is no action to express oneself unless there is desire to express oneself, and there is no communicative action unless there is a desire to communicate. But according to Kant and Searle, and the position being outlined here, it is the normative which explains the others: in particular, there is no desire to p unless and because one sees that one should p. In this picture, the normative is the explanatorily fundamental type of action, and the others (teleological, dramaturgical and communicative) are explained in terms of it.

If the felt urgency of desire can be expressed in L, can the hurtfulness of a pain also be expressed?

Consider the following sentence in L, with a free sentential variable X:

$\downarrow$DamagedElbow(me) $\rightarrow$ $\uparrow$See(X)

Now if this sentence is applied to itself, so that we substitute that very sentence for X, we get:

$\downarrow$DamagedElbow(me)) $\rightarrow$ $\uparrow$See($\downarrow$DamagedElbow(me) $\rightarrow$ $\uparrow$See($\downarrow$DamagedElbow(me) $\rightarrow$ $\uparrow$See($\downarrow$DamagedElbow(me) $\rightarrow$ …

This is a fixed-point. Having the pain just is seeing that the damaged elbow means I should notice it (where it = seeing that the damaged elbow means I should notice it).

Daniel Dennett has a similar understanding of the phenomenology of pain: "A less commonly recognized home remedy for pain is not to *distract*, but to *concentrate* one's attention on the pain. I discovered this for myself in the dentist's chair, thinking to take advantage of the occasion by performing a phenomenological investigation without the benefit of Novocain, and have since learned that this is a highly elaborated technique of Zen Buddhism. I recommend this enthusiastically. If you can make yourself study your pains (even quite intense pains) you will find, as it were, no room left to *mind* them: (they stop hurting) – though studying a pain (e.g. a headache) gets boring pretty fast, and as soon as you stop studying them, they come back and hurt. [*Why You Can't Make a Computer That Feels Pain*]

Here is a slogan for this general strategy of explaining phenomenological detail in terms of perception of the normative: feeling the urgency just is seeing the should.

21. *Comparison with Other Approaches*

This approach was inspired by the various writings of Garfinkel, Searle, Brandom, and the deontic logic literature.

This approach is like Garfinkel's in that it takes seriously the fact that the social practices which we take for granted are the result of a complex ongoing achievement. Because we are all (most of us, most of the time, anyway) so effortlessly good at participating in and co-creating these practices, the work required to maintain these practices is unnoticed by us, in the same way that we do not notice the work needed to maintain balance as we walk. But it is unlike Garfinkel's in that it is trying to create a formal recursive language in which to decompose the elements of social participation into a few simple building blocks.

This approach is like Searle's in that it is trying to find the building blocks out of which social practices are composed. It is also like Searle's in using normative judgments to explain desires, and not vice versa. It is also like Searle's in seeing status-function declarations as fundamental to the ongoing construction of social practice.

But this approach is unlike Searle's in that it involves a recursive structure in which there is an *indefinite* number of sentence forms, not just four types of direction of fit. The four directions of fit which Searle points out

are indeed central examples, but they are not the only examples. Another point of difference is that this approach takes as primary an individual's seeing, rather than a collective's seeing. In this respect, I side with Brandom and Garfinkel: the notion of "we" is not something that can be taken for granted in the beginning of the theory, but something that has to be *achieved*. This approach is like Brandom's in that it attempts to explain language in terms of practice. It is also like Brandom's in taking the notion of material inference and material incompatibility as prior to logical notions of entailment and negation. Like Brandom, it assumes an "I-Thou" conception of practice, rather than an "I-We" conception of practice.

But it is unlike Brandom's in that it begins with the idea that declarative word-to-world statements cannot on their own form an autonomous discursive practice, a language game you could play though you played no other. Brandom's GOGAR is a language solely involving ↓, whereas L is a language in which ↓ and ↑ are accorded equal status.

This approach is like deontic logic in that it is a formal language for representing norms. But it is unlike deontic logic in that it is fully *determinate*: there are no negations or disjunctions. It also differs from deontic logic in refusing to allow iterated embedding, like O(O(p)) – the only iterated embedding allowed is when ↓/↑ are alternated with See/Show.

22. *Summary*

I have introduced a formal (but pre-logical) language for describing what people do and should do when they participate in practices. I have shown examples of how L can be used to express rule-following, queuing, Garfinkel's summoning-phones, and sketched how it could be used to describe the Game of Giving and Asking for Reasons.

The explanandum was our capacity to participate in an indefinite variety of practices. The approach taken was to provide a combinatorial reduction of the indefinite variety of ways in which we can participate in practices to a few simple recursive constructs. But it is not a reduction from the normative to the non-normative. The explanans is irreducibly normative.

I am not, of course, claiming that people have sentences of L inside their head which they manipulate in order to understand and participate in practices. What I am saying is that L allows us to describe what people are doing and what they should do when they participate in practices. L describes the pattern – both what it is and what it should be. It follows that

if the agent were, *per impossibile*, to manipulate internal representations of sentences of L in an internalized language of practice, then he would be able to competently participate in our practices.

L is a language of practice, not a language of thought.

L is a language of practice, based on Kantian, Wittgenstinian and ethnomethodological considerations.

L is Kantian in that what it is to be an agent is to be committed to trying to see the truth. There is an inferential link from what is the case to what the agent should see:

$$P \rightarrow \uparrow See(x, P)$$

This schema has two instances. Agents should see what is the case, but they also need to see what should be the case:

$$\downarrow N \rightarrow \uparrow See(x, \downarrow N)$$
$$\uparrow N \rightarrow \uparrow See(x, \uparrow N)$$

L is Searlean and Wittgenstinian and ethnomethodological in that participants in a scene are continually responsible for manifesting the current state of the situation. The central insight is that the participants are continually accountable, and that they are continually giving the situation intelligibility by giving an account of their actions.

As Garfinkel puts it: "The ways in which the orderlinesses of the order of service are produced and managed are identical with the ways in which those orderlinesses are made accountable."

Making something accountable is *showing* that you *should* have done it. Garfinkel is saying that we produce the intelligibility of the social scene by showing the norms to each other. This central insight can itself be expressed directly in L:

$$\downarrow Do(x, a) \rightarrow \downarrow Accountable(x, a)$$

$$\downarrow Accountable(x, a) \rightarrow \uparrow Show(x, y, \uparrow Do(x,a))$$

Appendix A: Formal Definition of L

1. *The Syntax of L*

L is defined according to the rules:

States	$S ::= F \mid F(x_1, \ldots, x_n)$
Terms	$T ::= S \mid S.T \mid S{:}T$
Expressions	$E ::= T \mid See(x, J) \mid Show(x, y, J)$
Deontic Formulae	$D ::= {\uparrow}E \mid {\downarrow}E$
Conjunctions	$C ::= D \mid D \wedge C$
Judgments	$J ::= C \mid C \rightarrow C$

Here

- F is a function term, and x_1, …, x_n are terms referring to individual objects.
- S.T and S:T are ways of specifying sub-states of S, so we can build trees of expressions. In S.T, T is the unique child of S, in S:T, we are saying that T is one of the children of S, but there may be many others.
- $\uparrow$E says that E should be the case. $\downarrow$E says that E actually is the case.
- See is the term in L for scorekeeping the state of the practice. See(x, J) means that agent x sees that J. J itself may be either normative ($\uparrow$) or descriptive ($\downarrow$).
- Show is the term in L for manifesting the state of the practice. Show(x, y, J) means that agent x shows agent y that J. Again, J itself may be either normative or descriptive.

2. *Inference Rules for L*

The inference-rules come from a restricted version of input-output logic:

Weakening Output	From $A \rightarrow X$ and X implies Y, infer $A \rightarrow Y$
Transitivity	From $A \rightarrow X$ and $A \wedge X \rightarrow Y$, infer $A \rightarrow Y$
Conjunction	From $A \rightarrow X$ and $A \rightarrow Y$, infer $A \rightarrow X \wedge Y$

Output	
Equivalent Input	From A → X and A iff B, infer B → X

This is a restricted version of input-output logic, in that we expressly do not have the following inference rules:

Strengthening Input	From A → X and B implies A, infer B → X
Disjunction Output	From A → X, infer A → X ∨ Y

We don't allow Input Strengthening, because we want our arrow to represent *defeasible implication* – the sort of implication we use when we say if x is a bird, then x can fly (but if x is a bird and x is a penguin, then x can't fly). We don't allow disjunction output because disjunction is not an operator in L. There is no place for wishy-washy indeterminate claims like disjunction in a determinate language.

We also have the following axioms:

Generalized Throughput	↓A.B → ↓A ↑A.B → ↑A
See-to-Show	↑See(y, P) → ↑Show(x, y, P)
Show-to-See	↓Show(x, y, P) → ↓See(y, P)

We also have the following inference rules for See:

Modus Ponens	ɸ See(x, P) ∧ ɸ See(x, P→Q) → ɸ See(x, Q)
Transitivity	ɸ See(x, P→Q) ∧ ɸ See(x, Q→R) → ɸ See(x, P→R)
Conjunction Output	ɸ See(x, P→ Q) ∧ ɸ See(x, P → R) → ɸ See(x, P → Q ∧ R)
Generalized Throughput	ɸ See(x, ↓A.B → ↓A) ɸ See(x, ↑A.B → ↑A)

Where ɸ is ↑ or ↓ and P,Q, R are ↑E or ↓E for some expression E.

We also need axioms to express the (defeasible) two-way inferences between what is the case and what I should see.

I should see what is the case	P → ↑See(x, P)
If I should see it, then it is the case	↑See(x, P) → P

3. *Incompatibility in L*

One of the striking things about L is that instead of having a symbol for logical negation, it incorporates the determinate concept of *incompatibility*.

Two claims are incompatible in L if they have a common part which is post-fixed with "." but are different after the "."

Examples:

Compatible	*Incompatible*
X and Y	A.X and A.Y
F(a) and F(b)	H.F(a) and H.F(b)
A.B and A.B.C	A.B and A.C.D
A:B and A:C	A:B.C and A:B.D

X and Y are incompatible iff there exists an A, B, C such that X = A +.+ B and Y = A +.+ C and B ≠ C

3. *Incompatibility Resolution*

When using L in a real setting, the defeasible inferences involving → will issue in heaps of incompatible claims, which need to be resolved. There will be incompatibilities in both realms – in Up as well as in Down. How should we resolve between incompatible claims?

If X and Y are incompatible, we can choose to choose X over Y if our reasons for X are a superset of our reasons for Y.

Note that this only works because we are using defeasible implication which doesn't allow Input Strengthening.

Note that incompatibility resolution works just the same with ↑ and ↓.

4. *Extending*

To define the semantics of L, we need the concept of one term extending another. For example: A.B extends A.

X *extends* Y iff there exists a term Z such that X = Y.Z

5. *Semantics for L*

Define a *world* as a triple <Up, Down, Arrows>, where Up and Down are planes, and Arrows is a collection of arrows between the planes.
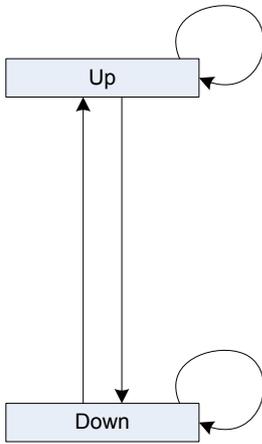
Define a *plane* as a triple <V, R, W> where

V : T → bool is a valuation of terms
R : Agent → World × World → bool is a relation on worlds, one for each agent
W is the world in which the plane resides.

Rules for how judgments are satisfied by worlds:

$w \leq \uparrow E$ iff $w.Up \leq E$
$w \leq \downarrow E$ iff $w.Down \leq E$
$w \leq X \rightarrow Y$ iff $X \rightarrow Y \in$ Arrows or $\exists Z. w \leq X \rightarrow Z$ and $w \leq Z \rightarrow Y$
$w \leq X \wedge Y$ iff $w \leq X$ and $w \leq Y$

Rules for how expressions are satisfied by planes:

$$p \le T \text{ iff } \exists \, R. \text{ Extends}(R, T) \text{ and } p.V(R)$$
$$\text{or } \exists \, Q \to R \in p.W.\text{Arrows such that } p \le Q$$
$$p \le \text{See}(x, P) \text{ iff } \forall \, w'. \, P.R_x(p.W, w') \to w' \le P$$

This form of semantics is related to input/output semantics in that the domain of interpretation itself contains a set of conditionals.

Note that we have no semantic rules for Show. Instead, Show is interpreted according to the inference-rules above:

See-to-Show	$\uparrow$See(y, P) $\to$ $\uparrow$Show(x, y, P)
Show-to-See	$\downarrow$Show(x, y, P) $\to$ $\downarrow$See(y, P)

6. *Syntactic Sugar: Local Negation*

It is useful to introduce some syntactic sugar to L. We introduce local negation to make the expressions simpler and more readable, but it is entirely inessential – it can be replaced without loss of expressive power.

Although I have been stressing that L does not contain an operator for negation, we can add a form of *local* negation within a term.

If we have a state T which we want to be able to negate, introduce two new terms T.True and T.False. Replace T with T.True and introduce ~T for T.False. This negated state is allowed only as the right-most state in a "." term. This allows us to write things like:

A.B.~C

But it does not allow us to write

~A.B.C

The reason for disallowing this sort of expression is that it is *indeterminate*: we do not know if ~A.B.C is true because we are actually in A.D, or because we are in A.B.E.

Appendix B: Describing Queuing in L

Recall that participants in a queue are continually seeing and showing their positions in line. They are continually preoccupied with place-work. In this section we express a simple queue in L.

In this example there is a distinguished object t which is what the participants are queuing to use. I will use x, y, z to range over agents. I will use ⊥ to denote the null object, so we can say After(x).Is(⊥) to express that there is nobody behind x. I will use ♣ to denote the null conjunction (which is always true).

Terms:

- Queue(t):Member(x).In – x is in the queue parameterized by object t. There is an incompatible term, Queue(t):Member(x).Out, which expresses that x is not in the queue.
- Queue(t):After(x).Is(y) — y is after x in the queue parameterized by object t. Note that there can only be one person in the queue after x because After(x).Is(y) and After(x).Is(z) are incompatible.
- Queue(t):Violation(x) – x has performed a violation
- StandBehind(x, y) — x is standing behind y. This relational term supports many different people standing behind x at once: StandBehind(x,y) is compatible with StandBehind(z,y)
- User(x).Is(y) – y is the user of x. Note again that User(x).Is(y) is incompatible with User(x).Is(z) – there is only one user of x at a time.
- Use(x, y) – y uses x. This relational term supports many different people using x at once.

Note that because Member, After and Violation are prefixed by a Queue(t) object, x can be in one queue while not being in another, and x can be behind y in one queue while simultaneously being in front of y in another queue. But to make the rules shorter and simpler, I will suppress the Queue(t) prefix in what follows.

Rules for Joining and Leaving the Queue:

1. ↑Use(t, x) ∧ ↓Member(x).Out → ↑After(x).Is(⊥)

If someone should use t and he is not already in the queue, then he should get to the back of the queue

2. ♣ → ↓Member(x).Out

By default, everyone starts off outside the queue

3. ♣ → ↓After(t).Is(⊥)

By default, the queue starts off empty

4. ↑ After(x).Is(⊥) ∧ ↓After(y).Is(z) → ↑See(x, ↓After(y).Is(z))

The person who has to get to the back of the queue has a responsibility to see where everybody is placed

5. ↑ After(x).Is(⊥) ∧ ↓After(y).Is(⊥) → ↑After(y).Is(x)

The person who has to get to the back should be placed behind the last person

6. ↑After(y).Is(x) → ↑Show(x,z, ↓After(y).Is(x))

If x needs to be behind y, then he must show others that he is behind y

7. ↑Show(x,z, ↓After(y).Is(x)) → ↑StandBehind(x, y)

If you need to show others that you are behind y, then you should stand behind y

8. ↓StandBehind(x, y) ∧ ↓After(y).Is(⊥) → ↓Member(x).In ∧ ↓After(y).Is(x) ∧ ↓After(x).Is(⊥)

If x stands behind y, and y was at the back, then x has joined the queue and he is after y, with nobody behind him

9. ↓StandBehind(x, y) → ↓After(y).Is(x)

If x stands behind y, then x is after y in the queue

10. ↓StandBehind(z, y) ∧ ↓After(y).Is(x) → ↓Violation(z)

If x is behind y, and z attempts to stand behind y also, then z has made a violation

11. $\downarrow$Violation(z) $\rightarrow$ $\uparrow$Rebuke(x, z) *Everybody should rebuke someone who commits a violation*
12. $\downarrow$After(x).Is(y) $\wedge$ $\downarrow$After(y).Is(z) $\wedge$ $\downarrow$WalkAway(y, t) $\rightarrow$ $\downarrow$After(x).Is(z) $\wedge$ $\downarrow$Member(y).Out *If y is between x and z, and y walks away, then z moves up to be behind x, and y is removed from the queue*

Rules for the person at the front of the queue:

13. $\downarrow$After(t).Is(x) $\rightarrow$ $\uparrow$User(t).Is(x) *If x is at the front of the queue, he should use the distinguished object*
14. $\downarrow$After(t).Is(x) $\wedge$ $\downarrow$~User(t).Is(x) $\rightarrow$ $\downarrow$Violation(x) *If x is at the front of the queue, and fails to use the distinguished object, he has made a violation*
15 $\downarrow$After(t).Is(x) $\wedge$ $\downarrow$User(t).Is(y) $\rightarrow$ $\downarrow$Violation(z) *If x is next in line to use the distinguished object and somebody else uses it, he has made a violation*

Notes:
- There will be cases when two different people y and z will both try to stand behind x, and the practice should have ways of resolving to decide who is actually behind x. Civilized queuing involves adding resolution rules like, for example:
- $\downarrow$StandBehind(y, x) $\wedge$ $\downarrow$StandBehind(z, x) $\wedge$ $\downarrow$Before(y,z) $\rightarrow$ $\downarrow$In(y) $\wedge$ $\downarrow$After(x).Is(y)
- These rules have universally quantified variables, but L does not contain variables! These rules are our generic representation of items which generate the token instances, and it is the token instances which are part of L.

In this simple example, the queue starts off completely empty with nobody using the distinguished object, t. Then x comes along, and uses t:

A. ↓After(t).Is(⊥)   *The queue starts off completely empty*
B. ↓Member(x).Out ∧ ↓Member(y).Out ∧ ↓Member(z).Out   *Each of x, y, and z starts off outside the queue*
C. ↑Use(t, x)   *x needs to use t*
D. ↑After(x).Is(⊥)   *(from A, B, C using 1) x should get to the back of the queue*
E. ↑See(x, ↓After(t).Is(⊥))   *(from A, D using 4) x should realize that t is at the end of the queue*
F. ↑After(t).Is(x)   *(from A, D using 5) x should be after t*
G. ↑StandBehind(x, t)   *(from F using 6 and then 7) x should stand behind t*
H. ↓StandBehind(x, t)   *Let us suppose that x sees that he should stand behind, and does it*
I. ↓Member(x).In ∧ ↓After(t).Is(x) ∧ ↓After(x).Is(⊥)   *(from H using 8) x has successfully placed himself at the back of the queue*
J. ↑User(t).Is(x)   *(from I using 13) x should use the distinguished object t*
K. ↓User(t).Is(x)   *Let us suppose that x now uses t*
L. ↑Use(t, y)

Now let us suppose that y needs to use t
M. ↑After(y).Is(⊥)   *(from I, L, C using 1) y should get to the back of the queue*
N. ↑See(y, ↓After(t).Is(x)) ∧ ↑See(y, ↓After(x).Is(⊥))   *(from M, I using 4) y should realize that x is after t and x is at the back of the queue*
O. ↑After(x).Is(y)   *(from I using 5) y should be after x*
P. ↑StandBehind(y, x)   *(from O using 6 and then 7) y should stand behind x*

Appendix C: Describing Rule-Following in L

Consider Wittgenstein's case of the pupil who is trying to understand a simple number series: adding 2. He sees the sequence of numbers 996, 998, 1000, and at some point – hopefully – he sees that he can go on.

Wittgenstein makes three fundamental points about this case:

- The pupil's understanding is not a private mental process
- It is *particular circumstances* that justify his claim I can go on
- Think of "Now I can go on" as a *signal*

We have been using the first two points throughout this paper. In this section I want to focus on the third.

"Now I can go on" is a way of showing the other that you see what you should continue the series with:

$$\uparrow\text{Show}(x, y, \downarrow\text{See}(x, \downarrow\text{Series}(\{996, 998, 1000\}) \rightarrow \uparrow\text{ContinueSeries}(x, \{996, 998, 1000\}, n))) \rightarrow \uparrow\text{Say}(x, y, \text{"NowICanGoOn"})$$

In other words: x's reason for saying I can go on is that he needs to show y that he sees that he should continue the series with n.

Here:

- Series(list) is a series consisting of the list of numbers
- ContinueSeries(agent, list, next) is the agent continuing the list of numbers with the number next
- nx is the number agent x thinks he should continue the series with

Saying "Now I can go on" itself has normative import:

$$\downarrow\text{Say}(x, y, \text{"NowICanGoOn"}) \wedge \downarrow\text{RequestsToContinue}(y, x, \{996, 998, 1000\}) \rightarrow \uparrow\text{See}(x, \downarrow\text{Series}(\{996, 998, 1000\}) \rightarrow \uparrow\text{ContinueSeries}(x, \{996, 998, 1000\}, n))$$

In other words, someone who claims that he can go on *has committed himself to* seeing how he should go on, if requested to do so.

Even simple examples like the pupil learning to add 2 are rich in alternating embeddings of See/Show and ↑/↓. L was designed to be the *simplest possible language* in which to express such constructs.

Bibliography

Boella, van der Torre. 2003. *Permissions and Obligations in Hierarchical Normative Systems*. Proceedings of International Conference on Artificial Intelligence and Law 03, Edinburgh

Boella, van der Torre, Verhagen (eds). 2007. *Normative Multi-agent Systems*. Dagstuhl Seminar Proceedings.

Brandom, R. 1994. *Making It Explicit*. Harvard University Press

Brandom, R. 2008. *Between Saying and Doing*, Oxford University Press

Dummett, M. 1977. *Can Analytic Philosophy be Systematic, and Ought It to Be?* Proceedings of the 1975 Stuttgart Hegel Conference

Garfinkel, H. 2002. *Ethnomethodology's Program*. Rowman & Littlefield

Garfinkel, H. 1991. *Studies in Ethnomethodology*. Polity

Makinson, D and van der Torre. 2003. *Input/Output Logics* Journal of Philosophical Logic 29 (2000) pp.383-408

Makinson, D and van der Torre. 2003. *Permission from an Input/output Perspective* Journal of Philosophical Logic 32 (2003) pp.391-416

Prakken, H (ed). 1999. *Norms, Logic and Information Systems*. IOS Press

Sacks, H. 1992. *Lectures on Conversation*. Blackwell Publishing

Searle, J. 1985. *Foundations of Illocutionary Logic*. Cambridge University Press

Searle, J. 1995. *The Construction of Social Reality*. The Free Press

Wittgenstein, L. 1953. *Philosophical Investigations*. Blackwell

Wittgenstein, L. 1983. *Remarks on the Foundations of Mathematics*. The MIT Press

von Wright, G.H. 1951. *Deontic Logic*. Mind 60

von Wright, G.H. 1953. *An Essay in Modal Logic*. New York Humanities Press