

Re-Expressing Normative Pragmatism in the Medium of Computation

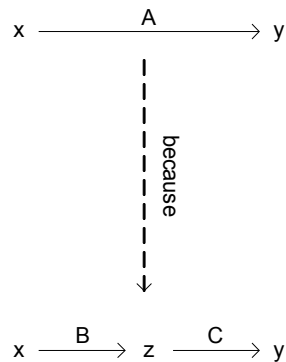
Richard Evans

Introduction

The central claim of normative pragmatism is that intentional states can be explained in terms of participation in practices. My aim in this paper is not so much to defend this claim as to rearticulate it in a different medium: the medium of computation. I describe two computer programs in which this claim is re-expressed. The first is the latest version of *THE SIMS*, in which participation in practices enables the Sims to do and understand more. The second is a prototype simulation of philosophical debate. In this second program, normative pragmatism is expressed at two different levels: once as the theory powering the implementation, and once as a set of claims which are debated by the simulated philosophers. Here, a philosophical theory is used to implement a system in which that very same theory is articulated, challenged, and justified.

Mediation

Some relations can be illuminated by splitting them in two, and positing a mediating entity between the relata:



If a relation can be understood in this way, then it is understood via *mediation*. The explanation is of the form:

$$\begin{array}{l} (\forall x) (\forall y) \\ \quad A(x,y) \\ \quad \text{because} \\ \quad (\exists z) B(x, z) \text{ and } C(z, y) \end{array}$$

Here, z is the mediator.

Some examples:

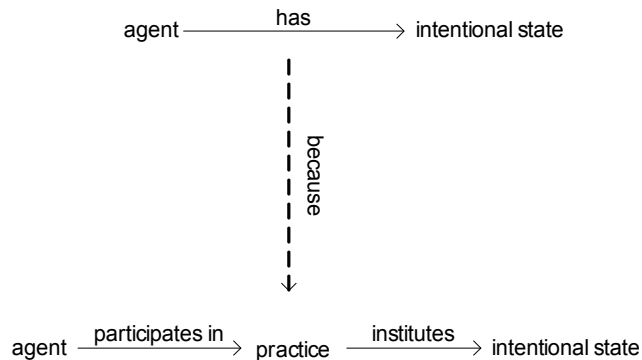
- x is the aunt of y because there is a person z who is the sibling of x and the parent of y
- x is a logical (proof-theoretic) consequence of y iff there is a proof z such that z starts with y and ends with x
- x is married to y because there has been a wedding ceremony z in which x participated in z and y participated in z

Normative Pragmatism as Mediation

One of the central claims in *Making It Explicit* is that intentional states should be explained in terms of participation in practices:

Expressions come to mean what they mean by being used as they are in practice, and intentional states and attributes have the contents they do in virtue of the role they play in the behavioral economy of those to whom they are attributed. [Brandom, 1994: 134]

This is another example of a mediating explanation. Having an intentional state is a relation between an agent and an intentional state which is best understood via a mediating entity - a social practice:



This fits the general pattern for mediating explanations:

$(\forall a:\text{Agent})(\forall s:\text{IntentionalState})$
Has(a, s)
because
 $(\exists p:\text{Practice})$ ParticipatesIn(a, p) and Institutes(p, s)

There is always, as it were, something between the agent and the thought: the practice. Much of the *Investigations* is taken up with diffusing potential counterexamples to this claim. “Even if many intentional states can be explained in terms of practices”, we might object, “surely some intentional states – such as understanding, reading, and hoping - are

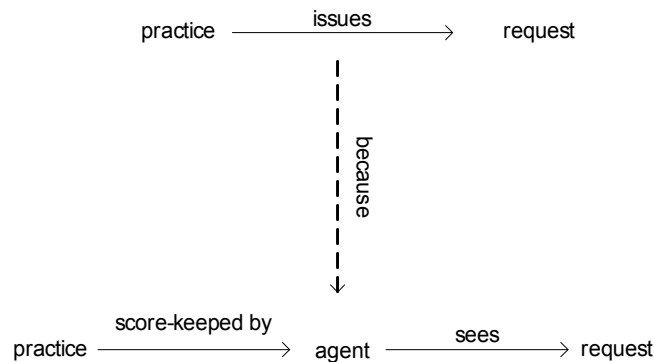
purely individual, and do not depend on participation in a practice?” Wittgenstein argues patiently that, even in these unpromising cases, the intentional state in question only makes sense within a practice (understanding §584ff, reading §156ff, hoping §584ff).

What a Practice Is

To understand normative pragmatism, we need to be more specific about what a practice is, and how agents relate to practices.

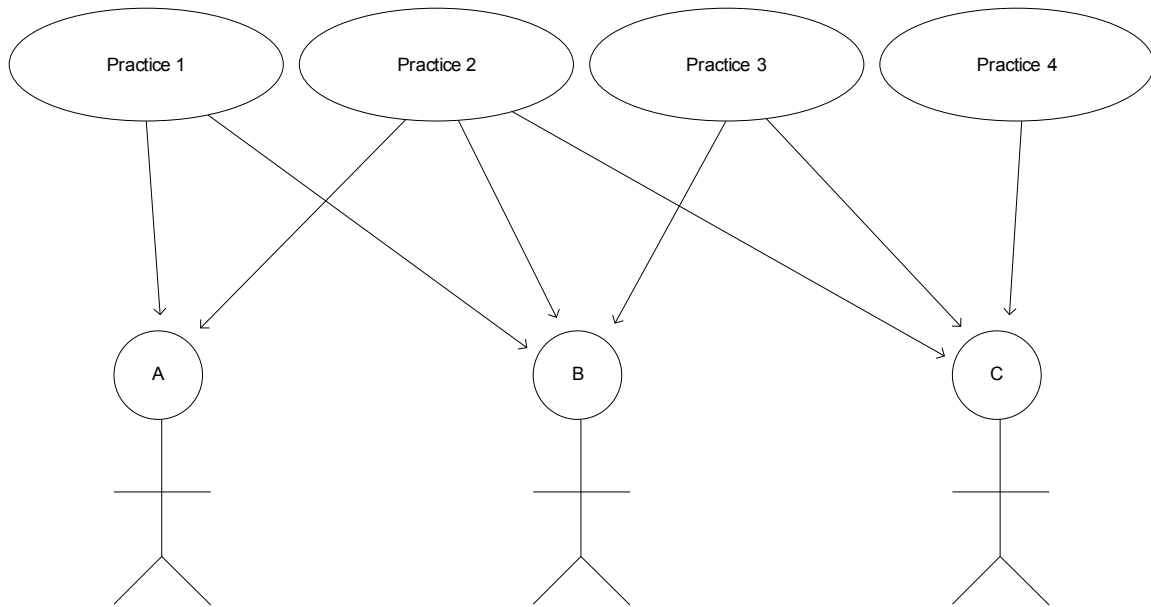
A practice is the most specific class which contains both games and languages as sub-classes: a “language-game”, where the hyphen is interpreted as a *range* (as in “A-Z”). A practice contains what agents should or may do in various situations. It issues different requests in different situations, can notice when norms are respected and violated, and respond accordingly (by issuing other requests).

This formulation unfortunately attributes agent-actions to practices: I talk of practices *issuing* requests, and *noticing* when norms are respected or violated. But I do not want to attribute agent-like mentality to these practices. This attribution is just an unfortunate side-effect of trying to be too quick, of trying to describe a practice in isolation, without describing an agent’s score-keeping that practice. A better way of putting the point about expectations is: when an agent keeps score on a practice, he sees that various things are expected of him and others, and he notices when those expectations are satisfied or confounded. When he sees that an expectation is confounded, he sees that other expectations become available. This is a mediating explanation of practices issuing requests:



How Agents Relate to Practices

An agent can participate in many practices at once. The various practices will compete for his attention. The expectations of a practice do not force the agent to comply. There is no direct causal link from the practice requesting something to its being done. Instead, the agent’s decision-making mediates between the request and action. He chooses between the various expectations from the various practices which are currently in play. Here is a stick-man diagram to illustrate how agents relate to practices:



Here we can see that each agent is in many practices simultaneously and that some practices contain only one agent.

Re-Expressing Normative Pragmatism in the Medium of Computation

Philosophers typically articulate their claims in natural language, and leave it at that. But sometimes we can get a deeper understanding of a proposition by seeing it from a different perspective, by seeing it re-expressed in a different medium.

I have found it helpful to re-articulate normative pragmatism in the medium of computation. But why computation, in particular, rather than clay, or interpretive dance?

Computers force us to clarify our thoughts. They admit no waffling or vagueness. Hand-waving is greeted with a compilation error, and a promissory note is returned, unread. Re-expressing normative pragmatism in the medium of computation forces us to get the details right. The original insights of normative pragmatism were first articulated in the difficult language of German Idealism. Recently, they have been rearticulated in the language of analytic philosophy by Wittgenstein, Sellars and Brandom. This rearticulation is progress, not just because the language of analytic philosophy is more accessible than Hegel, but because it is more specific. *Making It Explicit* is a towering example of specificity. But it is not the end of the road specificity-wise: a rearticulation of normative pragmatism in the language of computer science is even more specific still.

Computers are interactive and (unlike books) rich with counterfactual possibilities. When a claim is implemented within a computer program, we can test a variety of cases. By re-expressing a claim in an interactive medium, we get to understand the claim in a different way - by *playing* with it, rather than just *contemplating* it.

If we believe a claim, it should have an effect on our action. Sellars divides inferences (in the broad sense) into three types: language-language moves (where I infer one proposition from another), language-entry moves (where I arrive at a new commitment via perception), and language-exit moves (where I perform an action as a result of a belief). A proposition that is entirely disconnected from language-exits is *idle*. If we believe

normative pragmatism, and this proposition isn't idle, we should actually *do something* with this claim. One way of acting on the claim is by letting it guide our choice of AI architecture.

In the rest of this paper, I will discuss two examples of normative pragmatism made flesh: modeling social practices in THE SIMS, and a model of philosophical debate in a prototype called *Sim Philosophers*.

Modeling Social Practices in THE SIMS

In previous versions of THE SIMS, the Sims had no understanding of social norms. For example: my Sim had asked another Sim to come over to his house. But after he had invited her in, my Sim just walked off to have a bath - leaving his guest entirely on her own! This faulty behavior arose because the Sim did not understand that, after having invited his guest, there was an expectation to look after her. He violated the norm because he was unaware of it. For the latest iteration of THE SIMS, we have given the Sims a crude understanding of social norms, and have found this has made a real difference to their perceived believability.

How Practices are Modeled

A social practice is modeled as a hierarchy of clusters of expectations. Norms do not come singly, as individual rules which can be learned separately. They come in *clusters*. For example, the¹ social practice of Visiting between a guest and a host, involves a number of expectations:

- The host should look after the guest
- The guest should not over-step the bounds of hospitality
- The host may ask the guest to leave if he does over-step the bounds of hospitality

A practice is a tree of clusters of expectations. A practice can issue different requests in different states, notice when requests are performed, and change state accordingly.

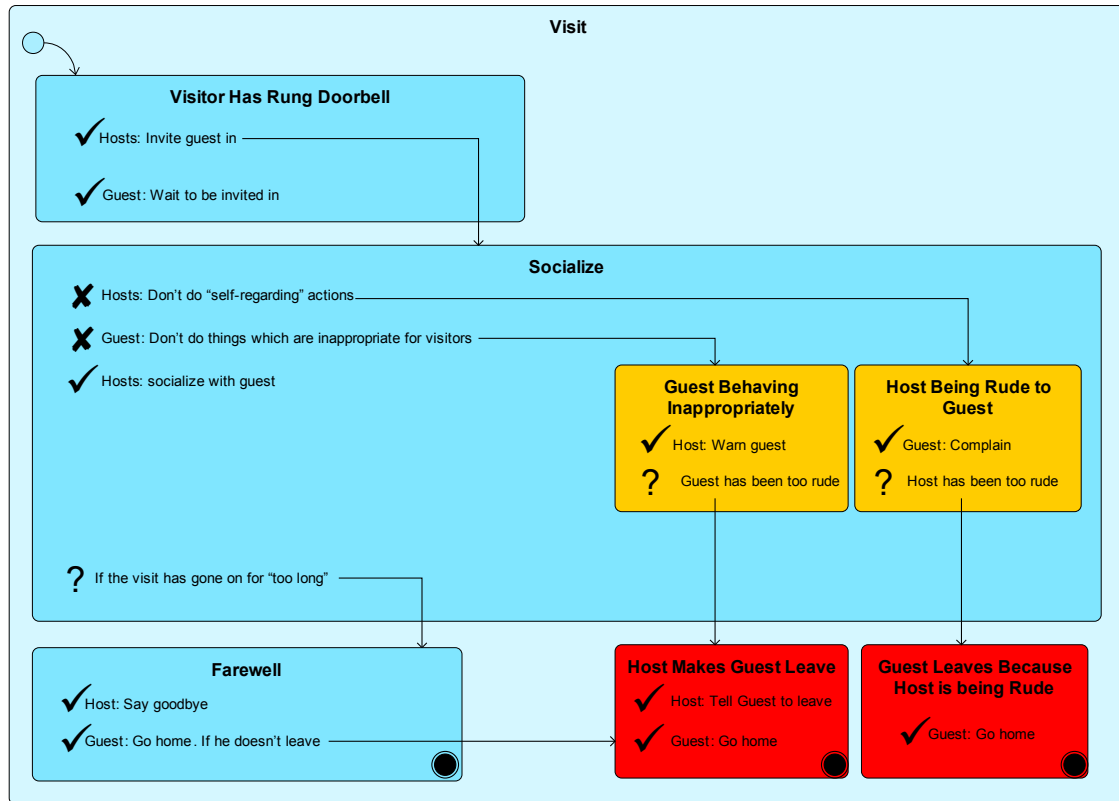
An agent can be in many social practices at once. For example, in the latest version of THE SIMS, if you invite your boss over for dinner, you and your boss are in (at least) three practices at once: Visiting, Mealtime, and Boss-Talking-With-Subordinate. Each of these practices has its own expectations. Sometimes these expectations can conflict.

At any moment, the Sim is faced with many different requests from many practices. He must decide which request to act upon. He understands each request as satisfying a subset of his desires, and chooses the request which best satisfies his current desires.

Some Examples of Practices from THE SIMS

¹ Of course it is a simplifying assumption to speak of one unique Visiting practice. There are lots of different types of Visiting practice for different cultures and sub-cultures, with differing expectations.

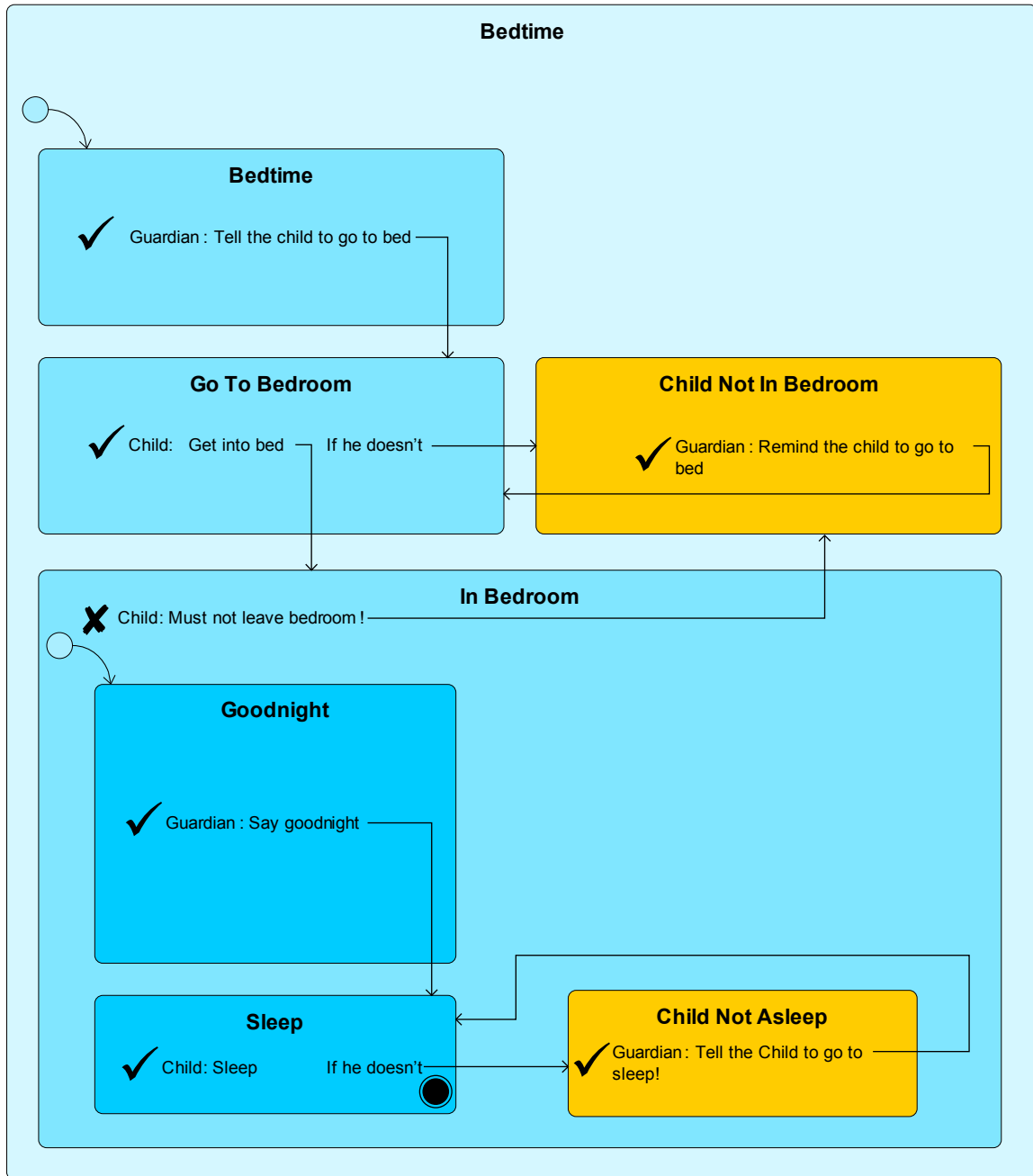
Practices in THE SIMS are described in UML (the Unified Modeling Language²) before being expressed in code. Here is a simplified version of the Visiting practice:



The Visiting practice handles various normal situations (Ringing the Doorbell, Socializing, Farewell), as well as various situations for handling irregularities (e.g. Guest Behaving Inappropriately). The practice can discourage certain actions (marked by ticks), and encourage others (marked by crosses). It can detect when norms have been violated and respond accordingly, by changing state (marked by an arrow) and thereby issuing new requests.

² Rumbaugh, Jacobson, Booch, *The Unified Modeling Language Reference Manual*.

Here is a simplified version of the Bedtime practice:



Note how the practice issues requests, and detects when those requests are violated. It handles violation by changing state and issuing new requests to fix the irregularity.

We are now familiar with the idea that many practices are constitutive rather than merely regulative: they don't just rule-out certain behavior, but enable new actions. For example, you can utter the sounds of the words "I do" whenever you like, but you can only

get married by saying “I do” within the social practice of the marriage ceremony. Some concrete examples from our latest version of THE SIMS:

- You can only *ignore someone* who is expecting to be talked to
- You can only *tell off a child for staying up past his bedtime* within the context of a bedtime practice which establishes the notion of bedtime
- You can only *stand someone up* who is on a date

In each case, the italicized actions are only intelligible because of the social practice in which they are embedded. Being in social practices gives the Sims more to do. But as well as enabling new actions, the practices also enable new intentional states:

- The host *being annoyed with the guest* only makes sense within a practice which defines appropriate behavior for guests
- The child *knowing he should go to bed* only makes sense within a practice which produces expectations of bed-going
- The suitor *expecting his date to show up* only makes sense within a practice

It might be objected, at this point, that these pre-linguistic practices are insufficient to produce genuine intentional states – that real full-blooded intentional states require understanding of a language. If we decide (and it is merely a decision) to only call such states “intentional states” if the agent can use language to express them, then our attention will shift towards the second computational implementation, described later, in which the Game of Giving and Asking for Reasons is implemented as a practice, in which agents *can* express their intentional states in language.

Related Work

Moses and Tenenholz [Moses and Tenenholz 95] have also developed computational models of social practices. They define a normative system as a set of restrictions on available actions such that, within these restrictions, there is a guarantee that a certain desirable condition obtains. For example: the desirable condition might be that no cars hit each other, and the restriction might be that all cars drive on the left hand side of the road. Part of the power of their work is that, using a type of modal logic, they can *prove* that certain norms guarantee certain desirable properties. Their approach is different from ours in that:

- Their social norms are atomic and self-contained (rather than embedded within a hierarchy of clusters of norms)
- Their social norms are restrictive, rather than constitutive – their norms rule out certain actions rather than enabling new options

Limitations of the Current Model

Our implementation of social understanding in the latest version of THE SIMS certainly makes the computer characters more believable, but it has some obvious limitations.

The first limitation is that the Sims do not keep track of their own models of the social practices³ (for purely practical reasons to do with minimizing memory overhead). Instead, the global system keeps track of the state of each practice, and each agent “magically” receives this information. This means that in our current implementation, there is no possibility of divergence of understanding as to how the practice is going. (In the second computer implementation, *Sim Philosophers*, discussed below, the individual agents do keep score, and they can diverge in understanding).

The second limitation is that the Sims cannot discover new practices. In our current implementation, the set of social practices which the Sims understand is fixed at design-time. They will never learn a new practice while playing.

The third major limitation is that the Sims are heteronomous, not autonomous. The Sims decide which practice to act on by scoring the requests according to their desires, but they are, as it were, prisoners of their own desires: they cannot but act on them. There is no possibility of them evaluating one of their motives, finding it distasteful, and replacing it with another. They are trapped in their own desires.

Lessons Learned from a Computational Implementation of Practices

The main lesson I learned from implementing these practices is that some practices are more robust than others, and that robustness presupposes the Game of Giving and Asking for Reasons.

Initially, I considered practices which encouraged and discouraged highly specific actions. These practices individuate actions by body movement. I shall call such practices *rituals*.

One example of a ritual is a marriage ceremony: the bride and groom say “I do” – it isn’t ok for them to say something that is semantically equivalent to “I do”, like “I hereby accept and willingly agree to the conditions mentioned” – you have to say the *actual words* “I do”.

Suppose we implement being-a-maid as a ritual: the maid is encouraged to hand-wash dishes if they are dirty. But this means that when a new way of washing dishes is added – using the dishwasher – the practice has to be modified. The practice is not robust against the introduction of new actions to achieve the same effect.

These practices, in which encouraged and discouraged actions were individuated by body movements, turn out to be extremely fragile: if things did not go according to plan, the behavior is unmasked as robotic. Compare the *Sphex* wasp:

The *Sphex* wasps drop a paralyzed insect near the opening of the nest. Before taking provisions into the nest, the *Sphex* first inspects the nest, leaving the prey outside. During the wasp's inspection of the nest an

³ See David Lewis, Scorekeeping in a Language Game, and Robert Brandom, Making It Explicit

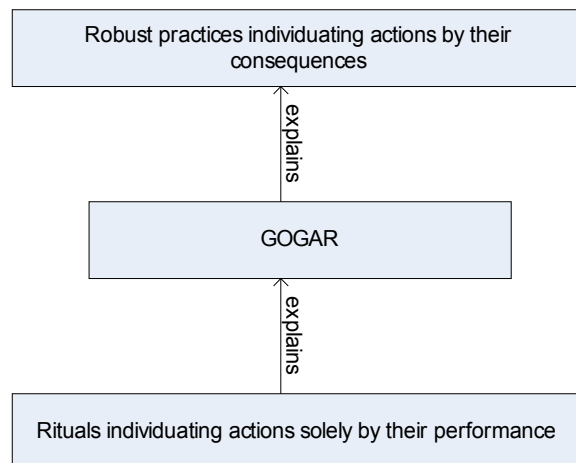
experimenter can move the prey a few inches away from the opening of the nest. When the *Sphex* emerges from the nest ready to drag in the prey, it finds the prey missing. The *Sphex* quickly locates the moved prey, but now its behavioral "program" has been reset. After dragging the prey back to the opening of the nest, once again the *Sphex* is compelled to inspect the nest, so the prey is again dropped and left outside during another stereotypical inspection of the nest. This [iteration](#) can be repeated again and again, with the *Sphex* never seeming to notice what is going on, never able to escape from its genetically-programmed sequence of behaviors⁴.

The wasp has a sequence of actions to achieve – if its plan is interrupted, it starts again from the very beginning– even if we can see that he was nearly there. It does not understand the practice in terms of intended consequences, but just in terms of a sequence of bodily actions, so it cannot begin again in the middle, for it does not see that the earlier actions' intent has been satisfied.

Making these practices more robust means defining norms in terms of actions *individuated by their consequences*. For example: the maid is encouraged to get the dishes clean – it is up to her how she does this. This extra level of indirection means the practice is robust against technological change, like the introduction of a dishwasher.

But understanding the consequences of an action requires an understanding of causal chains. Broad causal understanding requires linguistic understanding– either in that the agent himself formulates causal laws in language, or in that we use *our* linguistic formulation of causal laws to tell the computer agent the effects of the action. Either way, these complex practices presuppose one particular practice - the Game of Giving and Asking for Reasons (hereafter, **GOGAR**).

So we have two types of practice, rituals and robust practices. Ritual practices are simpler, but deficient. Robust practices individuate actions by consequence, and presuppose GOGAR. GOGAR is in the middle. Here is the first example of what I call a *normative sandwich*:



⁴ See Daniel Dennett, *Brainstorms* or http://en.wikipedia.org/wiki/Digger_wasp

The GOGAR is the practice which mediates between the robust practices, just as money is the commodity which mediates between the other commodities. This picture of GOGAR as central and fundamental to the higher practices is a core part of rationalism.

But if we acknowledge the central foundational role of GOGAR in a hierarchy of practices, then any computational re-expression of normative pragmatism must consider this crucial case. So next we turn to our second computational implementation of practices, an implementation of GOGAR.

To recap: ritual practices are primordial, but deficient. Practices which individuate actions in terms of consequence are more robust. But these robust practices require GOGAR (either because we use our understanding of GOGAR to give the agents a limited and brittle understanding of the consequences, or because we give them GOGAR themselves, so they can reason about causal consequences). We start to see that, as the rationalists have emphasized all along, GOGAR is central: it is the practice which mediates between the top-level practices, just as money mediates between the commodities. The next computer application is based on an implementation of GOGAR.

Sim Philosophers

Sim Philosophers is a prototype simulation of philosophical debate, albeit at a crude and superficial level. In each debate, there are various philosophers controlled by the computer, and one philosopher controlled by the player. The player can make claims, justify them, and challenge others. He can also just watch as the simulated philosophers argue amongst themselves. Each simulated philosopher has his own view of the debate, and these understandings can diverge. It relies on a computational implementation of the Game of Giving and Asking for Reasons (GOGAR), as described in Robert Brandom's *Making It Explicit*.

The telos of the GOGAR practice is to enforce the resolution of incompatible claims. An incompatibility cannot be left to stand – it must be resolved – either by justifying one of the incompatible claims, or by withdrawal. An agent is prompted to provide a justification for a claim if he has committed to it but is not entitled to it:

$$\begin{aligned} &(\forall a:\text{Agent}) (\forall p:\text{Sentence}) \\ &\quad \text{Should}(\text{Justify}(a,p)) \leftarrow \\ &\quad \text{Committed}(a, p) \wedge \sim\text{Entitled}(a, p) \end{aligned}$$

In *Sim Philosophers*, unlike THE SIMS, we model scorekeeping properly: each agent has his own version of the GOGAR and keeps track of the commitments and entitlements of all the speakers' claims. In this implementation, each person's version of the GOGAR agrees about the *commitments* attributed – there is an idealizing assumption that everyone can hear everyone else. But the different scorekeepers may well have very different views about the *entitlements* attributed, because they have different understandings of the material inferential relations between propositions. Scorekeeping GOGAR means keeping track of commitments and entitlements for each speaker. In this implementation, explicit commitments are tracked, but consequential commitments are computed only when needed, using lazy evaluation, rather than computing all consequences in advance.

The major challenge in implementing GOGAR is calculating entitlement correctly for each scorekeeper. Here, there are some significant differences from how entitlement is calculated in *Making It Explicit*.

The first difference is in how incompatibility is understood in terms of commitment and entitlement. Brandom understands incompatibility between p and q as when commitment to p precludes entitlement to q . But this formulation is insufficiently precise. Does he mean that the person doing the committing is the same as the person for whom the claim is entitled? Does he mean that the person doing the committing is the same as the person doing the scorekeeping? If $\text{Committed}_x(y, p)$ means that scorekeeper x attributes commitment that p to y , does he mean⁵:

1. $\text{Incompatible}_a(p,q)$ explained by: $\text{Committed}_a(a, p)$ precludes $\text{Entitled}_a(b, q)$
2. $\text{Incompatible}_a(p,q)$ explained by: $\text{Committed}_a(b, p)$ precludes $\text{Entitled}_a(b, q)$
3. $\text{Incompatible}_a(p,q)$ explained by: $\text{Committed}_a(b, p)$ precludes $\text{Entitled}_a(c, q)$

⁵ Here all variables are implicitly universally quantified.

The first is too weak. Because it insists that the scorekeeper is also the one doing the committing, it means that a scorekeeper will never acknowledge that his *own* claims lack entitlement. He may expect others to give reasons for their challenged claims, but he will never realize he needs to do so himself (unless he happens to make incompatible assertions on his own).

The second is too weak (in one way) because it focuses only on incompatibilities between the claims of one speaker. Incompatibility, if it is placed at the core of the game of giving and asking for reasons, has to be something that relates propositions from different speakers, prompting them to give reasons when their claims are challenged. It is also too strong (in another way): suppose that speaker S claims p and q, and p and q are incompatible (and he knows that they are). According to (2), S cannot be entitled to p or to q, *no matter what other claims he makes*. But suppose S now gives some very compelling reason, r, for q. Now according to (2), S is *still* not entitled to q, because of his commitment to p.

The third is too strong: it means that if two people make incompatible claims, neither of them can ever be entitled. It is an essential part of the game of giving and asking for reasons that entitlement can be *redeemed*. The third interpretation rules out this possibility of redemption, and means that an incompatibility removes entitlement forever, making the situation hopeless.

There are no other plausible ways to understand Brandom's claim that incompatibility is to be understood in terms of commitment precluding entitlement, so I use a different understanding of incompatibility, purely in terms of entitlement:

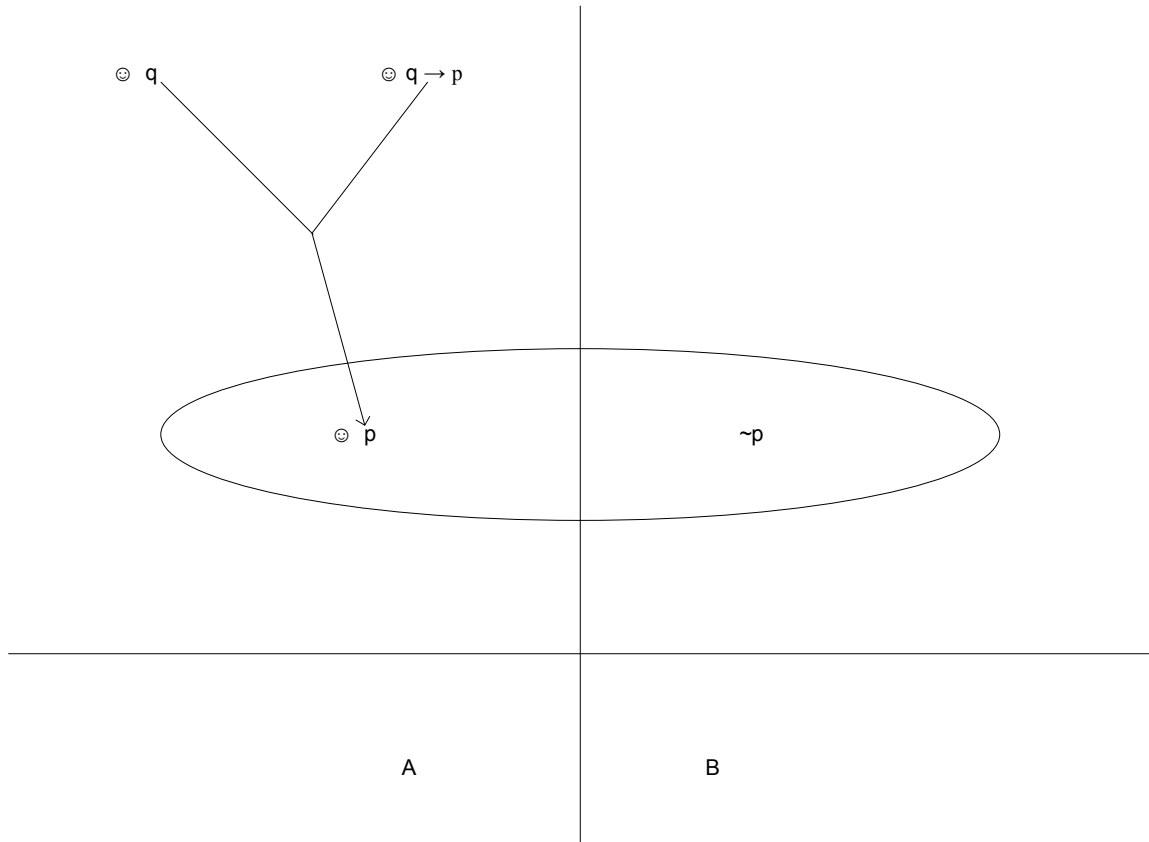
4. $\text{Incompatible}_a(p,q)$ explained by: $\text{Entitled}_a(b, p)$ precludes $\text{Entitled}_a(c, q)$

Incompatible propositions, in other words, cannot both be entitled. This is different from Brandom, who leaves this possibility open. In the penultimate sentence of section 4.5 of MIE chapter 3, Brandom says: "The public status of competing claims may remain equivocal in that neither the challenged nor the challenging claim can be vindicated successfully, *or in that both can be*". [My emphasis]

Intuitions about Entitlement

The hardest part of implementing GOGAR is calculating entitlement correctly for each scorekeeper. There were two main guidelines which constrained the entitlement-calculation rules. The first is that incompatible claims can never both be entitled. The second is that entitlement should be order-independent: the temporal order in which the claims are made should make no difference to the entitlement of those claims.

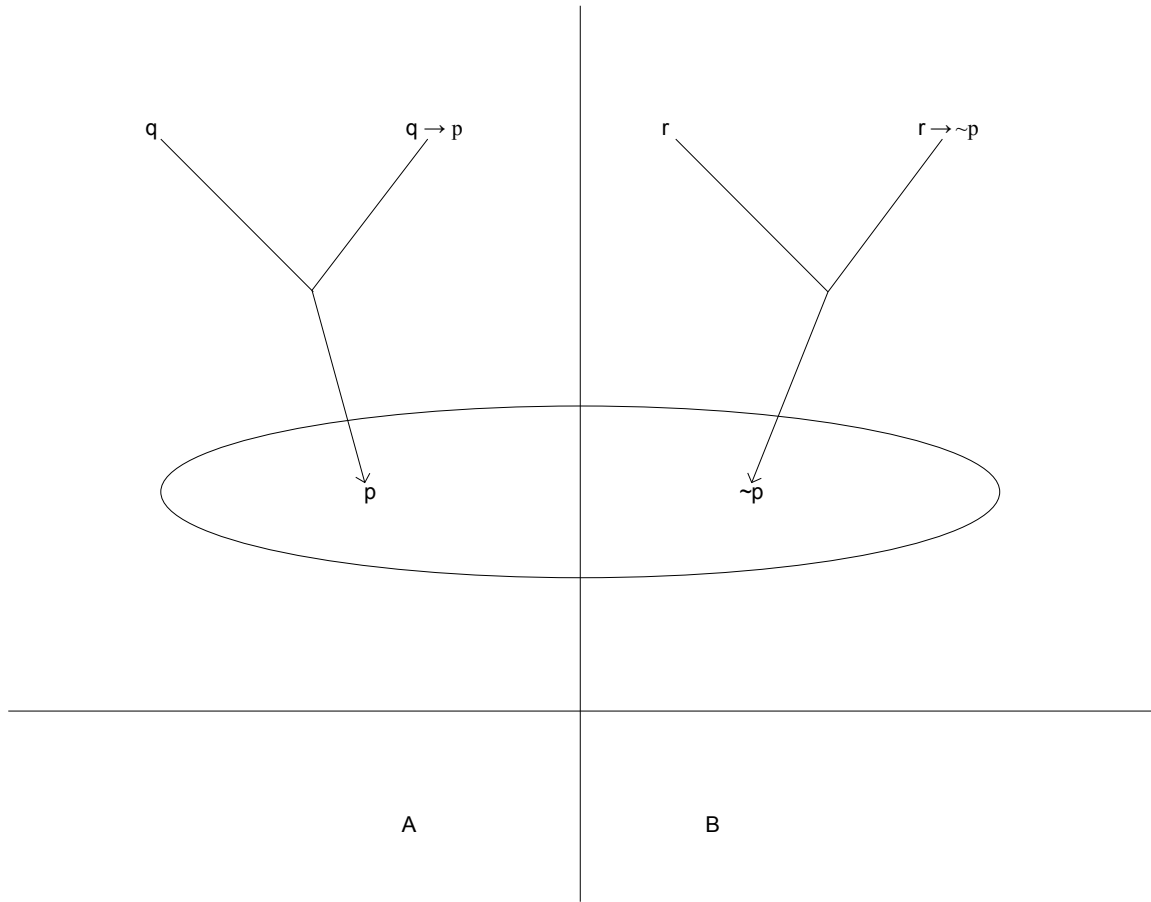
Before implementing the algorithm, I amassed an array of examples of simple debates, so that I had some test cases to check my algorithm against. I developed a simple way of representing debates as diagrams. The diagram includes the claims made by the various parties, the incompatibility relations, commitment-preserving relations, and assignments of entitlement. For example:



In this case,

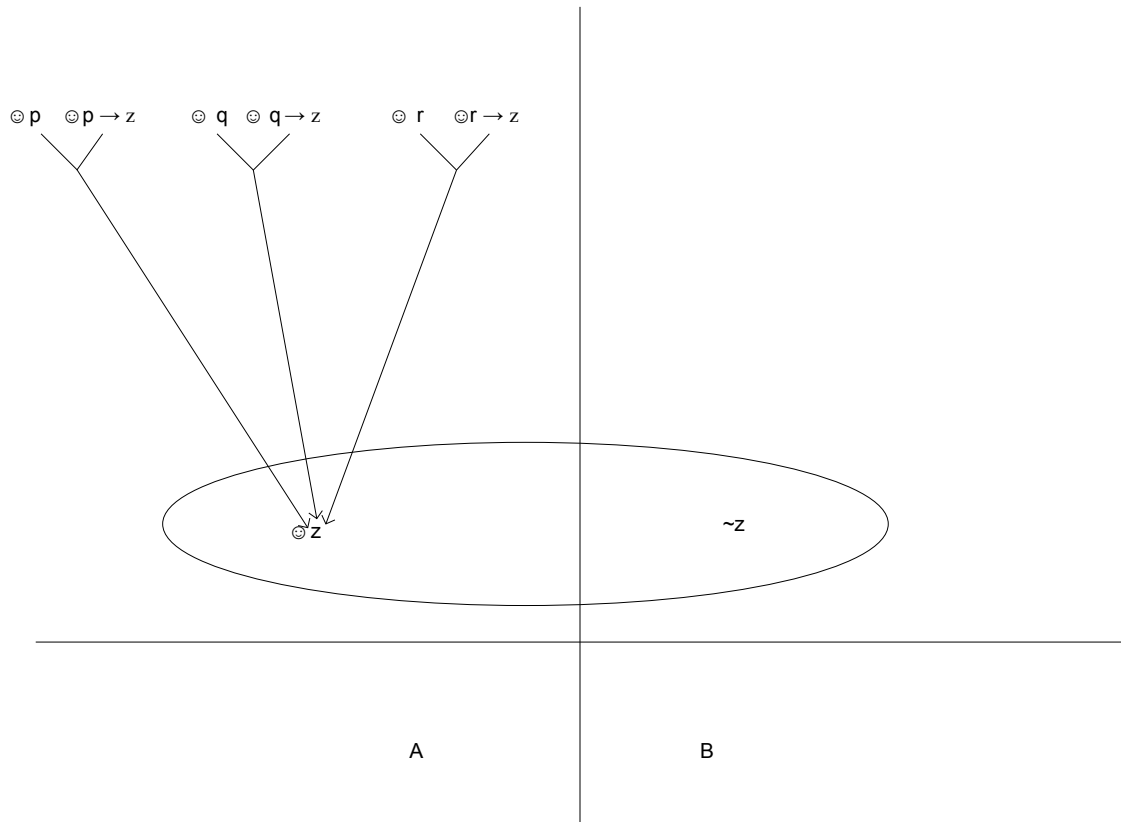
- The claims in the left column have been asserted by A, the claims in the right column have been asserted by B
- A has asserted p , q , and $q \rightarrow p$. B has asserted $\sim p$.
- p and $\sim p$ are incompatible (shown by a circle around both of them).
- q and $q \rightarrow p$ jointly commitment-entail p (shown by arrows)
- A is entitled to all his assertions (each assertion which is entitled has a smiley face ☺), but B is not entitled to $\sim p$

Consider a slightly more complex case:



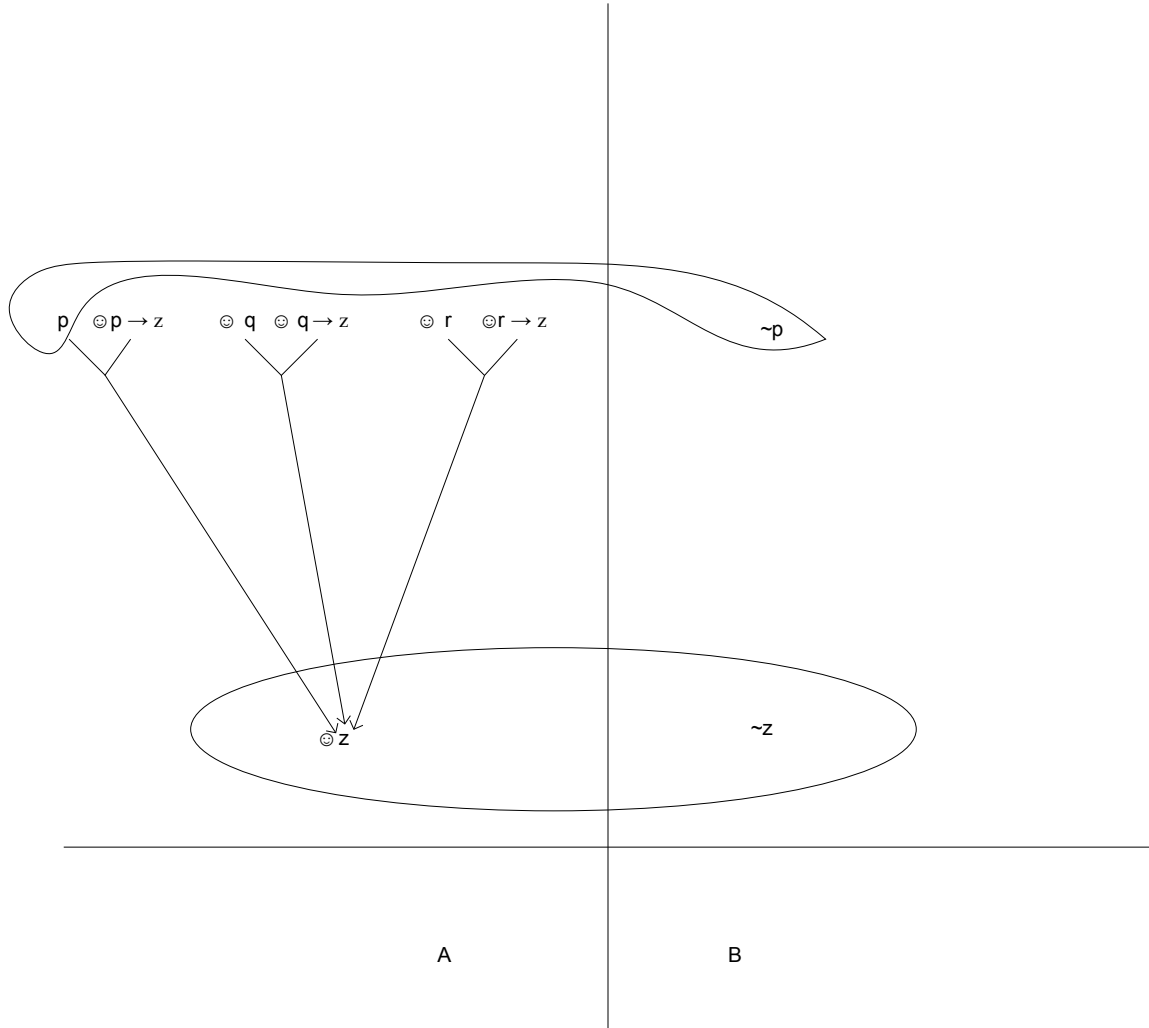
In this case, A and B have made incompatible claims (p and $\sim p$). They have given commitment-preserving reasons for each claim, but the reasons conflict. In this case, all these claims lack entitlement.

Now consider a case where A and B have made incompatible claims, and A has given multiple justifications for his claim:



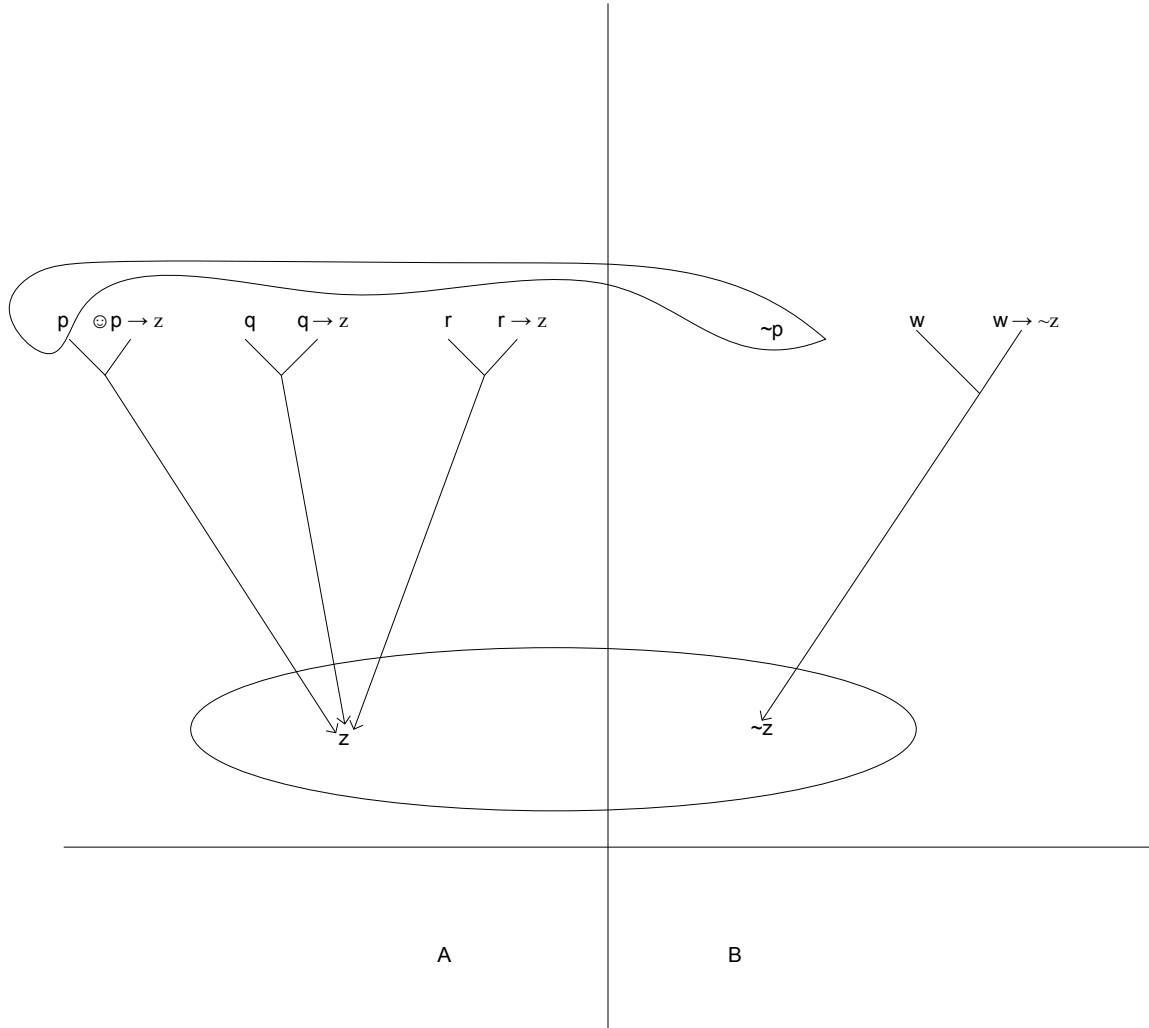
At this stage, A's claims are all entitled, whereas B's is not.

But now suppose B claims $\sim p$, thereby undermining one of A's reasons:



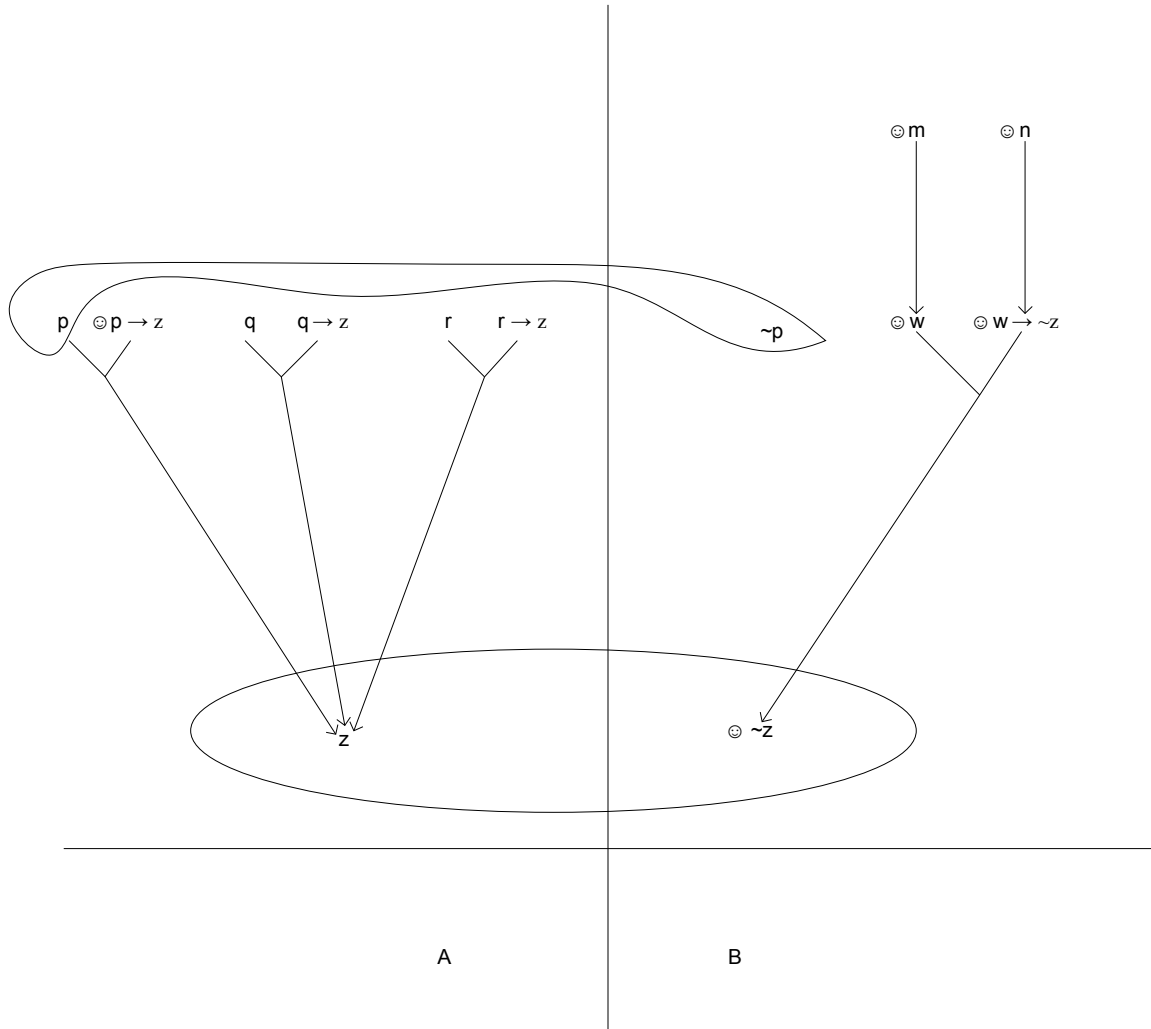
A has lost entitlement to p , since B has challenged it with $\sim p$. But notice that A's other reasons for z still stand, so A is still entitled to z .

Now suppose B gives a reason for $\sim z$:



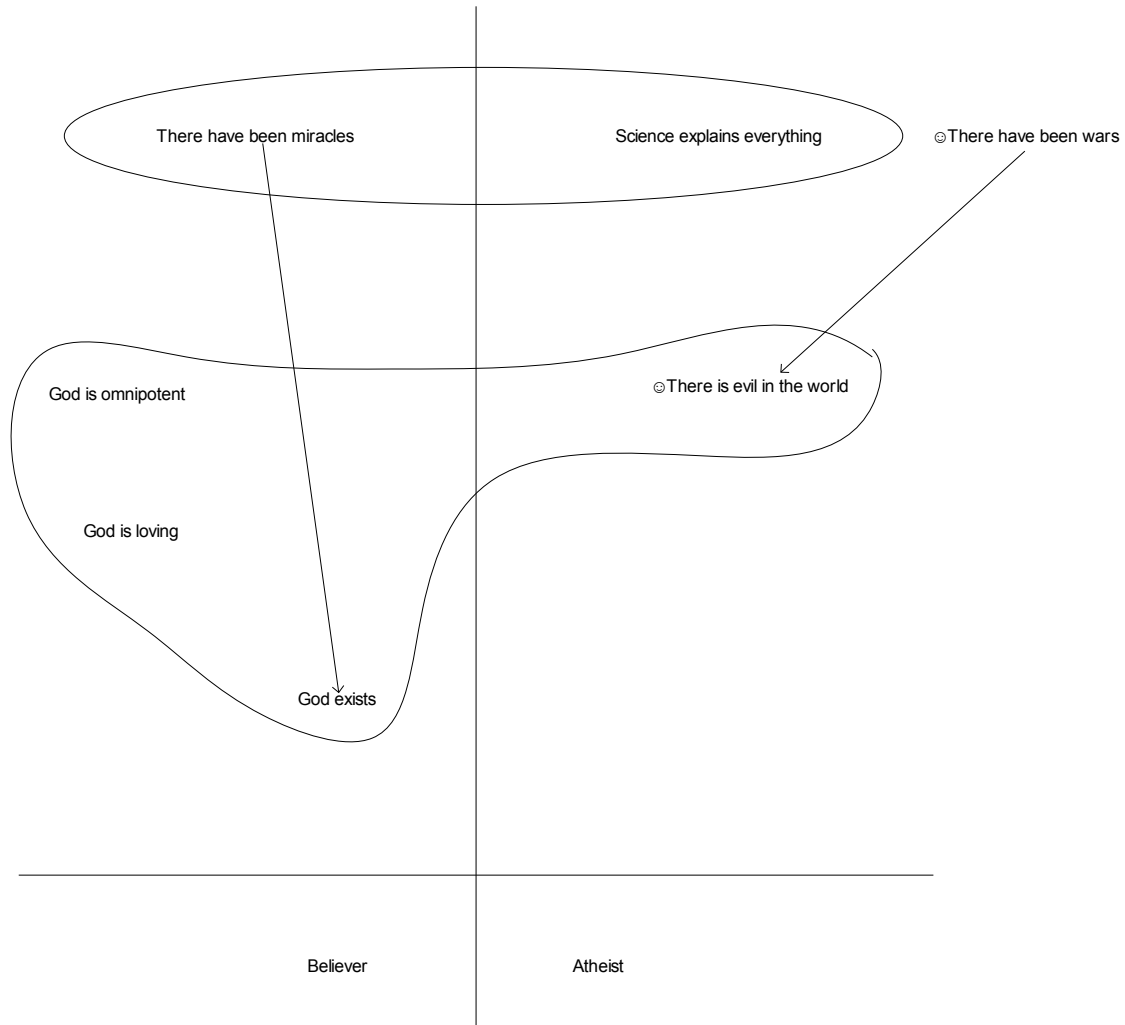
Now at this point, B has given just as much justification as A: they have both given a reason of depth 1 for their claim. Note that A has lost entitlement to almost all his claims. The fact that A has given three reasons whereas B has only given one makes no odds: it isn't the *number* of reasons that determines entitlement, but the *depth* of the reasons.

Suppose B now gave a deeper reason: a justification for w and another justification for $w \rightarrow \sim z$.



Notice that now B's claims are all entitled – the depth of the reasons determines entitlement, not the number of reasons at the same depth.

These examples so far have used propositional connectives. But I also considered a number of test cases using *material* inferences. For example:



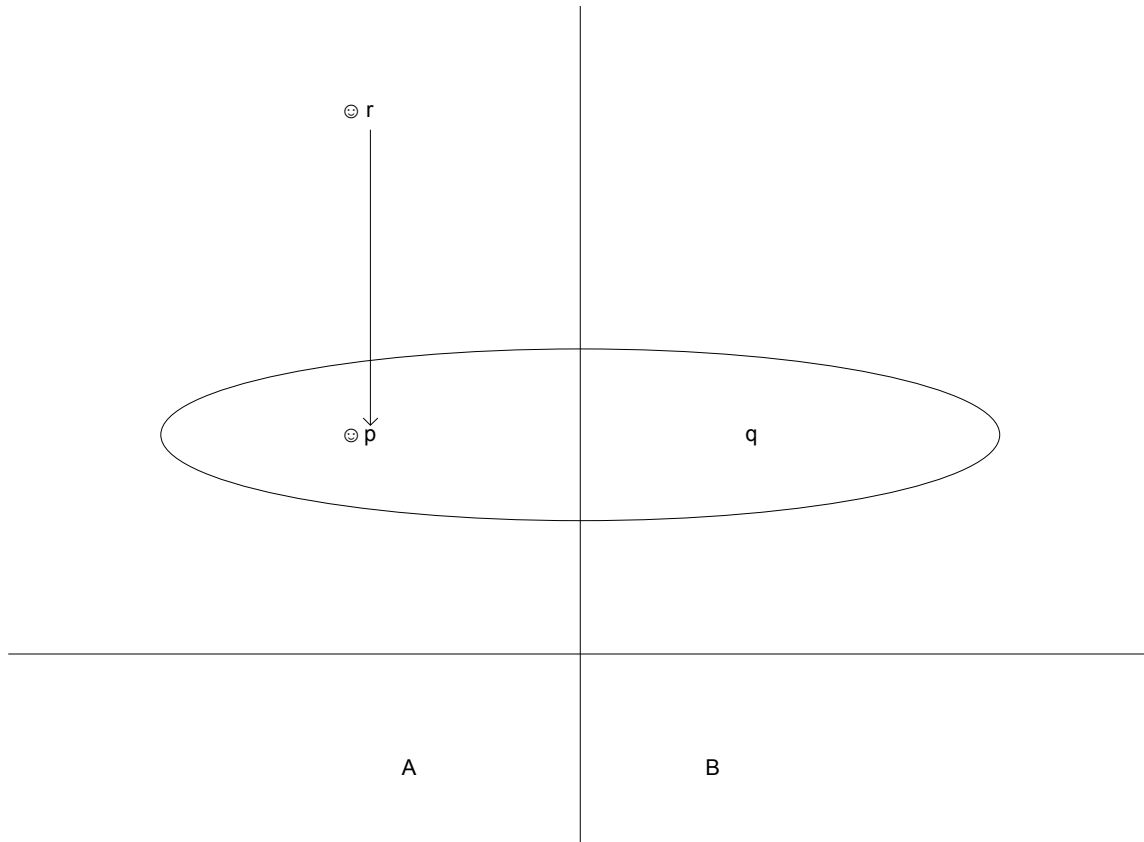
Updating Entitlement

The algorithm for calculating entitlement is a modified version of Brandom's procedure (in *Making It Explicit* 3.4.4):

- We start by assigning entitlement to *all*⁶ claims by default
- Then we remove entitlements based on incompatibilities
- Then we reinstate entitlements (based on commitment-preserving and entitlement-preserving consequences).

⁶ This is slightly different from Brandom, who assigns entitlement to claims based on the speaker's private beliefs. There is a *residual individualism* lurking in this part of his account.

The hard bit here is removing entitlements based on incompatibilities. Brandom glosses over some of the issues here, but re-expressing GOGAR computationally means that we have to get specific about the details. Incompatibilities need to be handled in a certain order. We can't just remove entitlement from every claim which features in an incompatibility set. For example, consider:



Here A has asserted p, B has asserted q (materially incompatible with p), and A has given a commitment-preserving reason for p, r. The commitment-preserving inference from r to p induces extra incompatibility sets: there are actually three incompatibility sets here: $\{p,q\}$, $\{q,r\}$, and $\{p,q,r\}$. If we removed entitlement from every claim in *each* incompatibility set, all three claims would lose their entitlement. But intuitively, p and q cancel each other's entitlement out, and then r comes along and revives p's entitlement.

So we want the incompatibility sets to be considered in a certain specific order: $\{p,q\} \leq \{q,r\} \leq \{p,q,r\}$. Once entitlement is removed from every claim in an incompatibility set, we remove the claims in that incompatibility set from the set of all candidate claims, and recompute the incompatibilities. In the example, once we have removed $\{p,q\}$, we are left with only $\{r\}$, which does not lose its entitlement. The underlying intuition is that a claim can only use its incompatibility power *once* before being used up⁷.

⁷ Compare Linear Logic, where individual propositions are resources which are consumed when they are used in inference.

Define a partial-order \leq on propositions, where $a \leq b$ if b is an entitlement-preserving reason for a . Extend this to a partial-order on sets of propositions in the natural way: $X \leq Y$ if every x in X is \leq some y in Y . For example, the reason why $\{p,q\} \leq \{q,r\}$ in the example above is because r is a commitment-preserving reason for p . To ensure we remove entitlement in the right order, we use the following method:

- Find all incompatible sets in the list of candidates
- Sort incompatible sets by \leq
- Remove entitlement from the most down-stream sets first
- Now remove all claims which have just lost their entitlement from the list of candidates
- Repeat until there are no more incompatibilities found in the candidates

How Agents are Modeled

As well as his version of the public GOGAR, each agent additionally has an *interiorized* GOGAR, representing his own beliefs. This is a separate instance of the same type of data-structure as the public GOGAR. The interiorized instance keeps track of the agent's beliefs which have not yet been asserted.

Because the private GOGAR stores his own beliefs, whereas the public GOGAR stores his public claims, the entitlement assignments in the two instances of GOGAR may differ. Agents are prompted to make private disagreements explicit as public disagreements: if X sees a claim by Y that has entitlement in the public GOGAR but lacks entitlement in his private GOGAR (because it is incompatible with one of X 's beliefs), X is prompted to assert that incompatible belief in the public GOGAR. He wants to make the entitlement assignments in the public GOGAR to be as close as possible to his private GOGAR. He wants, in other words, to make private incompatibilities public.

This brings us to another difference from Brandom. In Brandom's model, the scorekeeper uses his beliefs to assess whether the speaker has noninferential default entitlement to the claim. In this implementation, by contrast, *all* beliefs are entitled by default, irrespective of the beliefs of the scorekeeper. The interior GOGAR does not affect the entitlement-status of the public GOGAR until the interior beliefs are made explicit as assertions in the public GOGAR. I believe this is a minor fix to a residual individualism lurking in Brandom's account [Brandom 1994: Chapter 3.4.4 page 191].

At any stage during the debate, an agent may have a number of different types of speech act options available, and must choose between them. His options include:

- Making incompatibilities explicit by making a claim which is incompatible with one of the others' claims
- Giving a justification for one of my challenged claims
- Agreeing with someone who made a claim identical to one of my private beliefs
- Making a meta-claim about the status of the debate, to the effect that another speaker's claim is or isn't entitled

Each agent chooses between these options⁸.

How Philosophical Debates Are Modeled

Philosophical debates are modeled as collections of propositions, linked by inferential relations of material incompatibility, commitment-preserving and entitlement-preserving relations. Here is an excerpt from the program, when the Sim Philosophers were debating the possibility of AI:

Player: It is possible to build an intelligent machine.

Herbert Simon thinks that Player makes a very perceptive point that we can build machines as intelligent as us.

Hubert Dreyfus: Machines will never be able to understand the ordinary behavior which we take for granted in our everyday activity (thereby disagreeing with Player's claim that we can build machines as intelligent as us).

Herbert Simon thinks Hubert Dreyfus is wrong to claim that machines will never be able to understand the ordinary behavior which we take for granted in our everyday activity. Herbert Simon claims, by contrast, that we can build machines which are both intelligent and conscious.

Hubert Dreyfus: Machines can never hope to understand the intricate interconnected set of background practices which we continually take for granted and which make our everyday behavior intelligible (thereby disagreeing with Player's claim that we can build machines as intelligent as us).

Herbert Simon disagrees with Hubert Dreyfus's claim that machines will never be able to understand the ordinary behavior which we take for granted in our everyday activity. Herbert Simon asserts, by contrast, that we can build intelligent machines which will help us understand ourselves.

Herbert Simon: We should try to make intelligent machines (thereby disagreeing with Hubert Dreyfus's claim that machines will never be able to understand the ordinary behavior which we take for granted in our everyday activity).

Herbert Simon denies Hubert Dreyfus's assertion that machines will never be able to understand the ordinary behavior which we take for granted in our everyday activity and claims that we can make AI using GOFAI techniques.

Hubert Dreyfus disagrees with Herbert Simon's claim that we can make AI using GOFAI techniques. Hubert Dreyfus asserts, by contrast, that GOFAI is doomed to failure: there is no way to make an intelligent machine just by representing a whole heap of facts in a logical formalism and getting the computer to reason or plan over them.

Hubert Dreyfus: Know-that should be explained in terms of know-how, not the other way around. We cannot explain all basic capacities in terms of knowledge of explicit propositions.

⁸ At the moment I use a very simple static scoring function which always prefers making incompatibilities explicit over giving justification over agreement. I plan to use a better scoring function which takes into account personal relations with other speakers, one's current desire to be confrontational. This is the sort of scoring function that THE SIMS 3 uses to score speech acts.

Herbert Simon: We ought to try to understand ourselves.

Herbert Simon: It is only when you actually *build something* that you can test whether you really understand something.

Herbert Simon: Humans are just symbol-manipulating machines.

Hubert Dreyfus: GOFAI systems cannot understand ordinary everyday behavior (using a hammer to hit a nail, turning left at the left-turn).

Herbert Simon: We understand everyday objects by attributing predicates to them.

Hubert Dreyfus: We understand everyday objects (turning left at the left-turn) by participating in thick (object-involving) social practices (thereby disagreeing with Herbert Simon's claim that we understand everyday objects by attributing predicates to them).

Fearmonger denies Herbert Simon's assertion that we must try and built intelligent machines and claims that even if we could, it isn't a good idea to try to make artificial intelligences.

Lady Lovelace objects to Player's claim that we can build machines as intelligent as us and asserts, by contrast, that granted, machines can be very good at maths and symbolic reasoning, but they can never produce an original thought.

Roger Penrose objects to Player's claim that we can build machines as intelligent as us and asserts, by contrast, that it is not possible to build an intelligent machine.

Fearmonger: It is potentially dangerous to make AIs - they might kill us and take over the world!

Herbert Simon: The AIs we make will not be dangerous - we will ensure they are harmless when we build them (thereby disagreeing with Fearmonger's claim that it is potentially dangerous to make AIs).

Lady Lovelace: Machines can only do what they are programmed to do.

Roger Penrose denies Player's assertion that we can build machines as intelligent as us and claims that machines can never see the truth of their own Godel sentence.

I also used *Sim Philosophers* to debate normative pragmatism itself. First I took the core claims (both those that Brandom endorsed and denied) of Part 1 of *Making It Explicit*, and expressed them in First-Order Logic. Here are a few of the claims that were modeled:

Intentional States should be explained by norms

($\forall s$: IntentionalState)

($\exists n$: Norm)

($\forall a$: Agent)

Has(a, s) explained-by Under(a, n) and Fixes(n, s)

Regulism: Being under a norm means acknowledging an explicit rule

($\forall n$: Norm)

($\exists r$: Rule)

Captures(r, n) and

($\forall a$: Agent)
Under(a, n) explained-by ExplicitlyAcknowledges(a, r)

Pragmatism about norms

($\forall n$: Norm)
($\exists p$: Practice)
($\forall a$: Agent)
Under(a, n) explained-by In(a, p) and InsistsOn(p, n)

Regularism

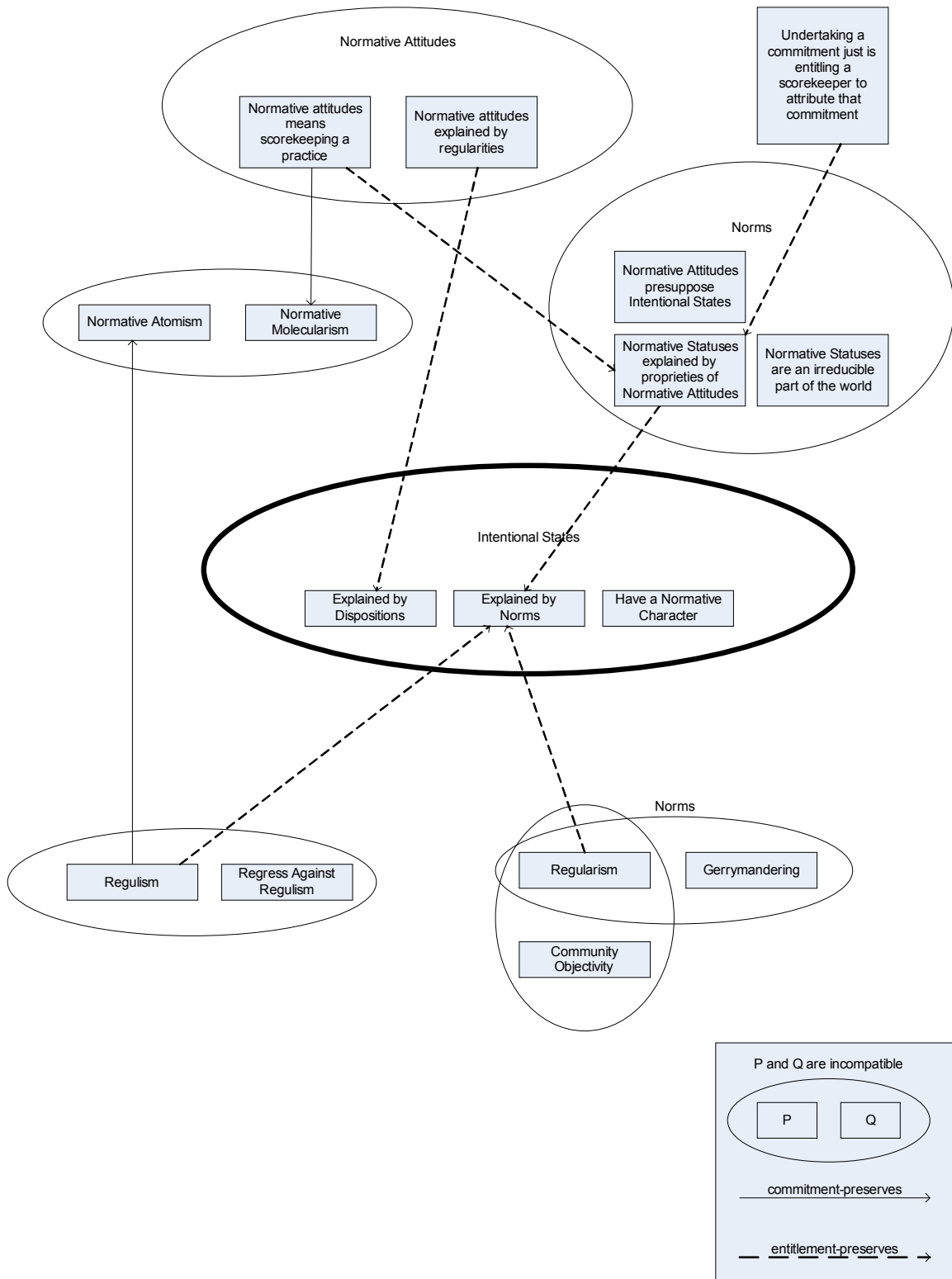
($\forall n$: Norm)
($\exists x$: Action)
($\forall a$: Agent)
Under(a, n) explained-by DisposedTo(a, x) and Satisfies(x, n)

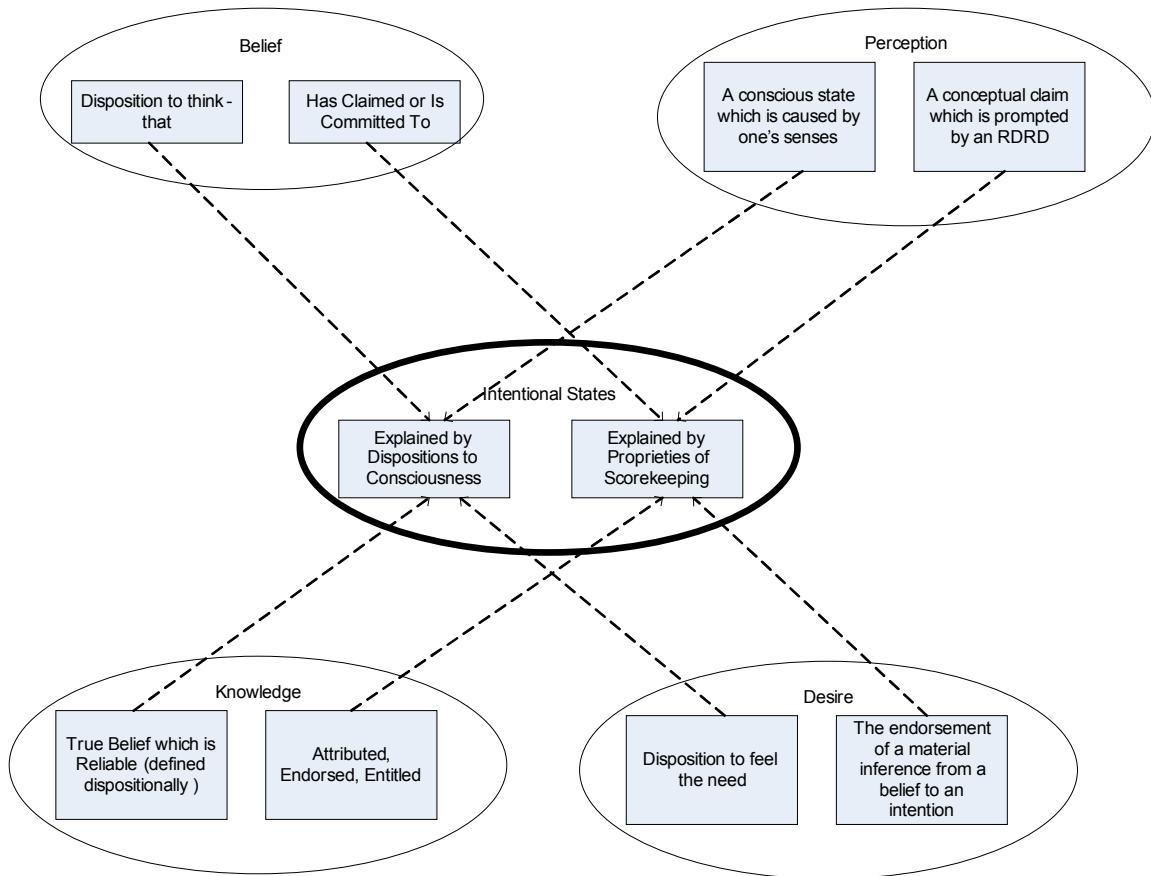
Our compulsion [to norms] is mediated by our attitudes

($\forall n$: Norm)
($\exists x$: Action)
($\forall a$: Agent)
Under(a, n) explained-by
InsistsOn(n, a, x)
AcknowledgesShould(a, a, x)

Then the core claims were re-expressed as propositions, together with inferential relations between them (material incompatibility, commitment-preserving and entitlement-preserving inferences).

Here is a part of the graph of material inferences:





Here is an excerpt from a recent run of the program:

Player: Intentional states are dispositional states; dispositions to have conscious experiences which capture their content.

Brandom disagrees with Player's claim that intentional states are dispositional. Brandom asserts, by contrast, that intentional states are just normative statuses.

McDowell thinks Player is wrong to claim that intentional states are dispositional. McDowell claims, by contrast, that intentional states are normative, but are not fully explained by their normativity.

Brandom: Normative statuses are explained by proprieties of normative attitudes; normative statuses are explained by how scorekeepers *should* keep score.

McDowell thinks Brandom is wrong to claim that normative statuses are explained by proprieties of normative attitudes. McDowell claims, by contrast, that normative statuses are irreducible.

Brandom: An agent's acknowledgement of the rightness of an action is explained by his scorekeeping a practice in which that action is mandated.

Brandom: Belief is explained as a normative status; belief that p is entitlement to attribute the claim or commitment that p.

Brandom: Undertaking a commitment just is doing something which entitles a scorekeeper to attribute that commitment.

Brandom: Scorekeeping can be explained by dispositions to sanction failure to comply.

Player: Being under a norm just means being disposed to do what it suggests.

McDowell: Objectivity requires the possibility that an agent consistently gets things wrong (thereby disagreeing with Player's claim that norms are just regularities).

McDowell denies Player's assertion that norms are just regularities and claims that the whole community can be wrong.

Brandom objects to Player's claim that norms are just regularities, and asserts, by contrast, that dispositions can be characterized in lots of different ways - there are always lots of possible patterns - no finite set of behavior will determine a unique norm.

And later:

Player: Scorekeeping practices involves keeping track in mental scoreboards of (linguistic) propositions representing the score.

Brandom disagrees with Player's claim that normative attitudes presuppose propositional intentionality. Brandom asserts, by contrast, that normative statuses are explained by proprieties of normative attitudes; normative statuses are explained by how scorekeepers *should* keep score.

Tradition finds himself in agreement with Player that normative attitudes presuppose propositional intentionality.

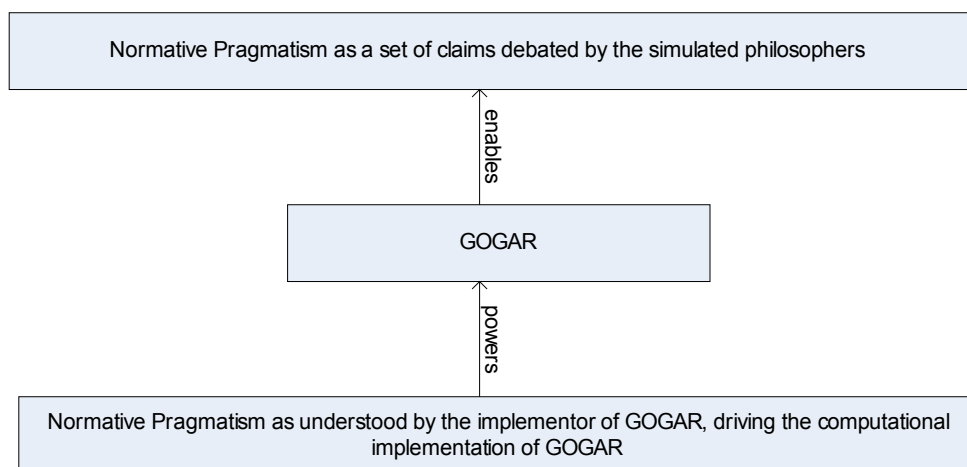
Onlooker: The claim that normative attitudes presuppose propositional intentionality has been challenged, and needs to be justified by Player.

Brandom: An agent's acknowledgement of the rightness of an action is explained by his scorekeeping a practice in which that action is mandated.

Kant thinks Brandom is wrong to claim that acknowledging a norm is scorekeeping a practice. Kant claims, by contrast, that acknowledging a norm is just having a propositional belief.

Haugeland thinks Kant is wrong to claim that acknowledging a norm is just having a propositional belief. Haugeland claims, by contrast, that acknowledging a norm is just having certain dispositions.

In this example, normative pragmatism is being re-expressed *twice*: once as the theory which powers the implementation of the GOGAR, and once as a set of claims within the GOGAR, which are debated by the computer characters.



Here we are using normative pragmatism to build a system which allows us to re-express that very theory again - this time explicitly - so that it can be challenged and justified. *This enables us to re-express philosophical thought in a new medium: interactive dialog.*

If we believe that significance emerges in a practice of enforcing the resolution of incompatibilities, then the monological essay cannot be the best way to represent a philosophical thought. Plato, of course, was well aware of this. But his dialogues are still static snapshots of a fictitious debate, cut and edited by the man behind one of the protagonists. *Sim Philosophers* provides a new way of expressing a philosophical view: expressing it within an interactive dialogue, rich with counterfactual possibilities, in which no particular observer is distinguished.

We have a particular philosophical theory - normative pragmatism - powering a computational implementation of an interactive debate, in which the very theory that was used as implementation can be articulated, challenged, and justified.

This is a dialogical variant of what Brandom (following Hegel) calls expressive completeness. Brandom's description of expressive completeness has a curiously (and uncharacteristically) monological slant: expressive completeness is when "the model reconstructs the basic resources needed to *describe* the theory itself"⁹ [my emphasis]. But if we prioritize the dialogical over the monological, we will not rest content with merely describing and explaining things. It isn't enough to build something which can *state* how it came to be – it must also be able to also *challenge* and *justify* that story.

Disclaimers and Limitations

When evaluating a piece of AI software which purports to implement some intentional action, we must be aware of the gap between our understanding of the action, embedded in the rich life-world of sophisticated adult humans going about their various affairs, and the implementation of the action in the computer. We must be honest to ourselves about this gap – failure to do so impedes progress. As Michael Mateas puts the point:

Every AI system consists of both a code machine and a rhetorical machine that interprets the operations of the code machine. The architectural surplus

⁹ Making It Explicit, p. 641

of the code system results from the embedding of code signs within broader, connotative meaning systems; words such as “goal”, “knowledge”, “embodiment”, and “emotion” simultaneously refer to specific technical configurations and operations, and aspects of human experience. The productive difference between the technical and human meaning of these terms results in technical innovation, new code machines, and new rhetorical constructions for narrating the code machine. The reduction of human subjecthood and experience that troubles these critics occurs when the terms in the rhetorical systems are *naturalized in their technical meanings*. When this happens, the productive difference between the technical and human uses of these terms is erased, doing violence to the original richness of the term, reducing its productiveness as a technical term, and contributing to rigidity in human notions of subjectivity. [Mateas, 2002, p. 195]

This paper you are reading is the rhetorical machine which is interpreting the operations of the computer software. When I say the simulated philosophers make philosophical claims, challenge others, and provide reasons for their claims, what is the relation between the claimings, challengings, and justifications in our rich multi-faceted life-world, and their counterparts in the machine?

A critic will claim that the simulated philosopher has, at best, only a superficial understanding of the philosophical ideas that are debated. But still, one might say in defense, one is reminded of the old saying about the elephant riding the bike: it isn't that it does it well, but that it does it at all. But the critic will reasonably object that the situation is significantly worse than that - it doesn't really do it *at all* – there's no real understanding here, just a pale imitation of it. There are, indeed, a number of fundamental limitations in the prototype as currently implemented.

One major limitation is that material inferences between philosophical propositions are only modeled at a rather coarse level. They understand material inferences between propositions, but there is currently no discerning of sub-sentential structure, no understanding that predicates or relations can be incompatible¹⁰. The philosophical concepts that are used (e.g. ‘perception’) are not connected to the language-game (e.g. of observation reports) which is their original home.

Further, the philosophical understanding given to the simulated philosophers is merely language-language moves (in Sellars' terminology): inferences between propositions and propositions. There are no language-entry moves (coming to believe a philosophical claim) or language-exit moves (they cannot act on a philosophical claim, as Marx, for example, insisted on).

Finally, all the philosophical beliefs and inferences are fixed at compile time: the simulated agents will never come to believe new claims or accept new inferences. This restriction applies equally to the player who is interacting with the system: the player cannot choose to say something that hasn't been anticipated up-front. The player isn't given freedom to type any philosophical claim he likes – he is restricted to a set of claims which were defined in advance by the implementer.

Once these limitations are acknowledged, I think it is clear that the simulated philosophers do not have non-derivative understanding of what they are saying. But being

¹⁰ John MacFarlane has an implementation of GOGAR which does model incompatibilities between predicates.

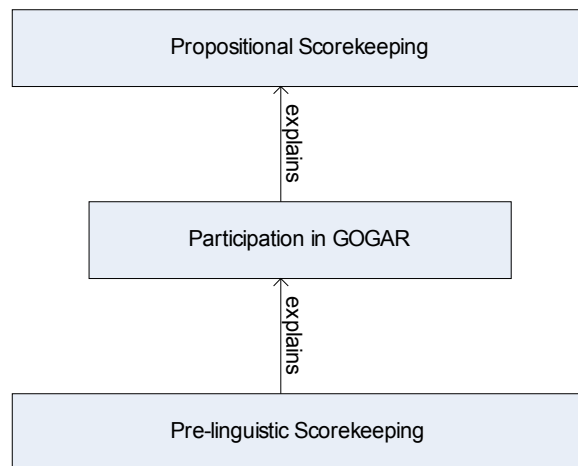
honest about the gap is the first step to bridging the gap. When we start to address some of these limitations by seeing them as missing features to be implemented - if we were, for example, to model sub-sentential structure, and provide language-entries and exits by unifying the SIMS architecture described earlier with the Sim Philosophers architecture above - then we would, I submit, have implemented agents with some (limited but non-derivative) understanding of the philosophical issues that they are discussing.

But the point of this is not to evaluate a particular piece of hastily-written software, but to draw philosophical lessons from it.

Lessons Learned

Two major points have emerged from implementing a computational model of GOGAR within a simulation of philosophical debate. The first is that calculating entitlement is rather different, and rather more complicated, than Brandom suggests in *Making It Explicit*. Incompatibility is a relation between entitlements, and not, as Brandom suggested, a relation between a commitment and an entitlement. Calculating entitlement is not as simple as removing entitlement from all incompatible propositions. The incompatible sets need to be sorted, so that entitlements can be removed in a very specific order. The second point is that when belief is handled as a separate, interiorized version of the GOGAR, we no longer need to use the agent's beliefs to evaluate whether or not someone is entitled in the public GOGAR. By cleanly separating the private GOGAR from the public, we are able to remove the residual individualism lurking in Brandom's account.

There is one more point which emerges clearly in the computational implementation. This is the distinction between two very different sorts of scorekeeping, which are sandwiched between GOGAR:



Pre-linguistic scorekeeping is the primitive capacity to keep score, even if we can't *say* what the score is. At the computational level, the score which the computational agent is keeping is just binary data. But participating in GOGAR makes it possible for the agents to also do *propositional* scorekeeping¹¹ – to express the state of practices in language. For

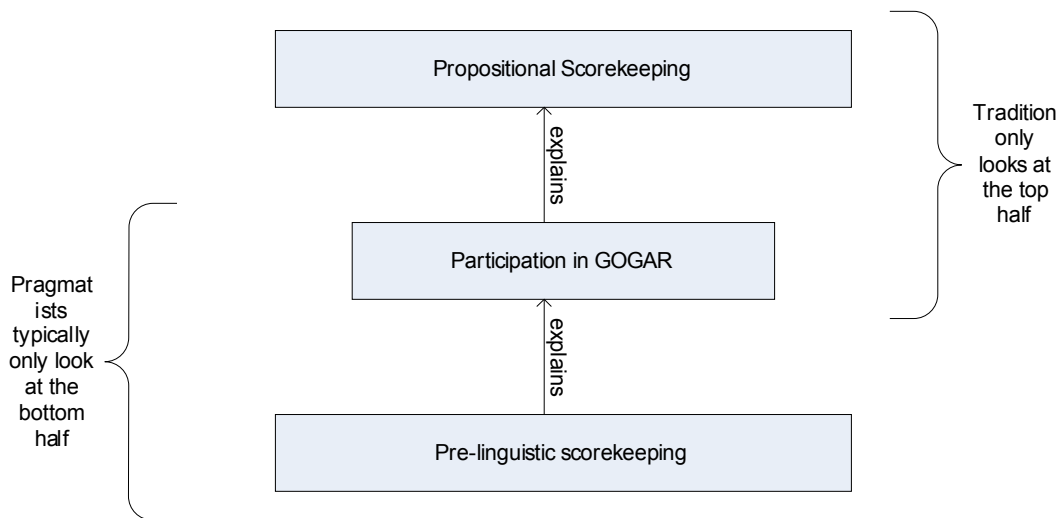
¹¹ Brandom describes propositional scorekeeping in *Making It Explicit*: “Once the expressive resources of a full range of semantically and pragmatically explicating vocabulary are in play, those who have mastered them can keep score explicitly, by making *claims* about each other's doxastic, practical and inferential

example: in *Sim Philosophers*, the agents can *say* when a claim is lacking entitlement and needs to be justified. In the sample debate described earlier, the onlooker noticed when the player’s claim was challenged:

Onlooker: The claim that normative attitudes presuppose propositional intentionality has been challenged, and needs to be justified by Player.

GOGAR is sandwiched between these two different sorts of scorekeeping: it is because we can (pre-linguistically) keep score that we can keep score on GOGAR, and it is because we can keep score on GOGAR that we can *say* explicitly what the score is.

If you only look at the top half of this normative sandwich, you can see why so many philosophers have seen linguistic competence underlying social practices. Just looking at the top half is what Heidegger called *burying-over*. “On the basis of their integrated structure in a system they present themselves as something ‘clear’ which is in no need of further justification” [Heidegger, *Being and Time: The Preliminary Concept of Phenomenology*]. GOGAR explains the complex practices, but covers up the underlying explanation - the pre-linguistic scorekeeping. This is why Normative Pragmatism is unintuitive or even unthinkable to so many – they are only looking at the top half of the sandwich.



If you only look at the bottom half of the sandwich, you can see why some pragmatists have de-emphasized the importance of GOGAR as itself a foundational capacity.

But when we look at the whole sandwich, we can see why the debate has oscillated unprofitably between the tradition and the pragmatists: they were both right, at different levels.

Conclusion

I have been trying to motivate a rearticulation of normative pragmatism in the medium of computation. This is the next phase in a progression of specificity: normative pragmatism

commitments” (p. 641)

was first expressed in the language of German idealism; recently it was re-expressed in the language of analytic philosophy; now it can be re-expressed once more in software. But the programs I have discussed represent only the beginnings of what can be said in this relatively new expressive medium.

Bibliography

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

Forbus and Wright. 2002. *Some notes on programming objects in The Sims™*.
http://www.qrg.northwestern.edu/papers/Files/Programming_Objects_in_The_Sims.pdf

Heidegger, M. 1927. *Being and Time*. HarperOne.

Lewis, D. 1983. *Scorekeeping in a Language Game* (In *Philosophical Papers Volume 1*).
Oxford University Press.

MacFarlane, J. 2006. *A computational implementation of GOGAR*.
<http://johnmacfarlane.net/gogar.html>

MacFarlane, J. 2006. *Pragmatism and Inferentialism*. Forthcoming.

Mateas, M. 2002. *Interactive Drama, Art and Artificial Intelligence*. Carnegie Mellon
University. <http://www.lcc.gatech.edu/~mateas/publications/CMU-CS-02-206.pdf>

Moses and Tennenholtz. 1995. *Artificial Social Systems*.
<http://citeseer.ist.psu.edu/cache/papers/cs/732/http:zSzzSziew3.technion.ac.il:8080zSz~moshetzSzcai.pdf/moses95artificial.pdf>

Rumbaugh, Jacobson and Booch. 2004. *The Unified Modeling Language Reference Manual*. Addison-Wesley.

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