On model reduction and equivalence in ABMs

Daniel Tang

July 5, 2024

1 Poisson models and Markov processes

Let a Poisson model be a dynamic system on some discrete state space such that the probability that the process will transition from state x to state x' in time dt is given by $\rho('x|x)dt$. If we let ρ_{ij} be a transition matrix such that

$$\rho_{ij} = \begin{cases} \rho(i|j) & \text{if } i \neq j \\ \sum_{k \neq j} -\rho(k|j) & \text{if } i = j \end{cases}$$

and let X be a vector representing a probability distribution over the state space, then we can define the Poisson model as a continuous time dynamic system

$$\frac{dX}{dt} = \rho X$$

so that

$$X(t) = e^{\rho t} X(0) = e^{(\mu - I)rt} X(0) = \sum_{k=0}^{\infty} \frac{(rt)^k e^{-rt}}{k!} \mu^k X(0)$$

where $r = \max_j \sum_{k \neq j} \rho(k|j)$ is a scalar and $\mu = \frac{\rho}{r} + I$. It can be seen that each entry of μ is non-negative and the sum of entries in each column is 1, so $\mu^k X(0)$ is the state of a discrete time Markov process at time k.

Since μrt commutes with Irt we can separate and expand to give

$$X(t) = e^{-rt} e^{\mu rt} X(0) = \sum_{k=0}^{\infty} \frac{(rt)^k e^{-rt}}{k!} \mu^k X(0)$$

which we recognize as a sum of powers of μ weighted with a Poisson distribution. So, each continuous-time Poisson model ρ has an associated discrete-time Markov process $\mu = \frac{\rho}{r} + I$. The two are related by the fact that, for a given start state distribution, the state of the Poisson model at time t is the weighted sum of states in a trajectory of the Markov process, where the weights are given by a Poisson distribution with rate rt.

This means, among other things, that as $t \to \infty$ the state distribution of a Poisson model tends to a uniform distribution over the attractor of its associated Markov model. So every Poisson model tends to a steady state distribution (i.e. a single point in distribution space).

2 Social norms and social agents

Let a *social role* be a tuple $\langle S, \sigma_0, A, Q \rangle$ where S is a domain of social states of the actor of which σ_0 is the default state given at the beginning of an episode, A is a domain of social actions that can be performed by the actor and Q is a social-quality function in $S \times A \to \mathcal{R}$.

Let a *social norm* be defined as a tuple $\langle r_1, r_2, \tau \rangle$ where r_1 and r_2 are social roles and τ is a state transition function in $S_1 \times A \times S_2 \to S_1 \times S_2$ saying that if two agents playing roles r_1 and r_2 are in states s_1 and s_2 respectively and agent 1 performs action a on agent 2 then, after the action the agents have state s_1' and s_2' respectively. [can this be separated by defining the social quality of actions *performed on other* given other's state? In this way, my behaviour towards you is constrained by your role, not mine. Can we define social state change solely by own actions? Or perhaps much simpler, just define the input alphabet of a social role]

Let a *social connection* be a tuple $\langle N, s_1, s_2 \rangle$ where N is a social norm, s_1 is the social state of the first agent in N and s_2 is the social state of the second.

Let a *social network* be a directed graph, where each edge is associated with a social connection. So, each agent in a social network can be thought of as being in a number of social states (one for each edge) and each social state comes with a constraint on behaviour in the form of the social Q-functions.

Let a *social agent* be a node in a social network. The agent's internal state is the set of social states on its edges. The agent's behaviour towards each agent it has a social connection with is constrained by the Q-function associted with that connection.

Agents in a social network need not, and often do not, physically exist. God, the government and Microsoft do not exist but many people model them as agents to which they have social connections. This raises no difficulty as long as the physical meaning of social actions involving the non-existant agent don't require its physical presence, and there's no reason it should need to. These actions can be instrumental in the coordination of behaviour between physically existant agents. I can be employed by, or sue Microsoft without ever expecting to meet face to face.

2.1 Social network as closure

If an agent models its world and assumes every other agent models the world in the same way, then evey agent in the model contains an instance of the model and we have an infinite regression, so we need some kind of closure.

If we assume that the state of the social network is public knowlede, through observation, gossip etc. (where X is public knowledge if everyone knows X and it's public knowlege that everyone knows X) then, under discounting, agents can model their world as consisting of agents that act according to the public knowledge. For example, agents can use the social norm as the off-tree approximator and run Monte-Carlo tree search to account for potential breaking of the social norm. In this way, agents are able to come to the same equilibrium.

2.2 Social network as abstraction

In many situations, we're only interested in the social actions that each agent performs, so we can model a set of agents as interacting via social actions. In reality, these social actions are grounded in physical actions and if we're interested in the details of this, then each social action must come with an extension which is a set of sequences of physical actions that qualify as this social action. The agent must then choose a physical action in consideration of the social interpretations of that action. In general, the physical substrate of the agent's social world adds additional constraints to the sequence of social actions it may perform.

Modelling at the physical level also allows agents to perform actions that have multiple social meanings, so a single physical action or sequence of actions can perform many social actions. It is also possible for a physical action to be unclassifiable as a social action yet state changing in the social norm (e.g. I sit down at a restaurant, the waiter brings me the menu, i set fire to it). In this case the norm is considered null and new norms need to be negotiated.

3 How do agents learn social norms?

[Language, telling stories. By accidentally breaking social norms and being told off. It seems implausible that we learn social norms only through personal experience and direct observation: I haven't learned not to murder people by murdering a few and finding that it wasn't really worth it, nor even by whatching people murder others and imagining that it isn't worth it. Rather, I learn it on the social level, as part of the culture in which I live.]

A learner of social norms that only has access to observed physical behaviours must somehow learn the abstraction function, the social quality function and the transition function. If thrown into a world where a social norm is already prevalent and stable, and we assume people are guided by social norms, then all this can theoretically be done by minimising the error in predicting other's behaviours.

Alternatively, can can abstract over the physical and assume agents make observations at the social level of abstraction. This glosses over the learning of the abstraction function, which we learn from gossip etc. and focusses on the social dynamics. If agents are able to make social observations of their surroundings and their interactions then an agent can construct a probabilistic policy for each social norm (or if the norm type is itself negotiated, and we have a universal state and action space, then this can be a single policy...although it's hard to see the advantage of this compared to stratification by norm).

4 how do agents negotiate roles to form new social connections?

Context. Verbally.

Again, we can abstract over the physical details of this and assume there is a public vocabulary of norms, and connections have an associated social norm type-id.

5 How do social norms evolve?

If agents choose to create social connections at a rate proportional to their expected reward then some social norms will be popular and others not. Ultimately, some may die out.

New norms can be created by more or less intelligent mutation of existing norms (perhaps even unintentional misunderstandings or chance events), or specialisation of general-purpose norms: [this assumes there's a heirarchy of abstraction among norms, which we haven't talked about yet...] for example, hunting large game and sharing the meat may emerge as a specialisation of you-help-me-and-i'll-help-you norm...?

5.1 Social judgement in social norms

In a purely Q-learning environment, punishment and praise after the fact is ineffective in one-shot games. However, if a social norm consists of a whole policy, and is agreed upon at the beginning of an interaction, then a credible threat of contingent punishment and/or praise can be instrumental in making the policy a Nash equilibrium.

Perhaps social judgement goes two ways: it rewards the actor for socially beneficial behaviour, but also signals to everyone which behaviours should be copied and which avoided. This makes sense if social judgement is outcome based and/or defined on an abstract level. That is, the ability to make social judements allows everyone to easily see whether a given behaviour is good or bad, but it is more difficult to generate instances of behaviours that are good.

5.2 Justice

It's clear that concepts such as justice are largely shared and preceed law (i.e. are more than a respect for the rule of law). It's also clear that humans don't define justice in terms of reward, but rather in terms of some kind of social value (Has justice been done if I give you two camels for your daughter?).

Moreover, a potential new social norm could conceivably conflict with other social norms (even though they have different state spaces), and so a new social behaviour cannot become established as a norm if it conflicts with already established norms. This implies that there is much more structure to social norms than we have allowed so far. What exactly is this structure, is it essential and how do we represent it?

Justice isn't a social norm as we've defined it so far in that it doesn't seem to have an action space. It is more a property of a story (sequence of interactions

between a set of agents). Unless it is a formula for how to take an unjust story and "serve justice" to transoform it into a just story.

Or perhaps a person performs an unjust act on another, and justice is the social norm of punishment of the actor and/or redress to the injured. The player performing the punishment/redress can be abstract (e.g. government). These are the norms. The concept of justice help us think about these norms.

Or perhaps better to think of justice as a concept that is part of the conceptual apparatus we use to reason *about* social norms, and perhaps to intellecutally explain our innate feelings.

6 Generative abstraction

To capture this, we propose *generative abstraction* (which is something like the inverse of a formal semantics of natural language: while a semantics goes from a sentence to an extension, an abstraction goes from a set of observations to a representation. The abstraction is generative in the sense that the representation space grows exponentially with the size of the concept space).

Suppose we have a complex dynamic system we wish to model. Let its state space be T. Suppose also that we have two abstract models with state spaces A and B, that are different abstractions of T. We can create a new, compound model with state space $A \times B$ by simply allowing the state of the system to be a pair of states: its A-state and its B-state. A simple approximate dynamics for the compound model can be formed by just using the dynamics of the corresponding sub-models to update the A and B states independently. A good example of this is to model the dynamics of a 2-dimensional projectile in a field by splitting into perpendicular x and y dimensions. In this case, the dynamics is exactly "factorizable" (i.e. no error is introduced).

If we interpret being in an abstract state as meaning that the target state is contained within the extension of the abstract state, and two abstract models are correct, then the simple dynamics of a compound model is also correct and the extension of the compound state is the intersection of the extensions of the constituent states. This is true for the componding of any number of models. Although some state combinations are impossible (the intersection of the extensions of the consitiuent states is empty), if we start in a possible state, we will never reach an impossible state if the constituent models are correct.

7 Generative social norms

If we allow an agent to simultaneously play multiple roles towards another agent, and it is public knowledge how the social Q functions of these roles combine, then the number of social Q functions that can be described increases exponentially with the size of the vocabulary of social norms. Note that social states do not have extensions, they describe a social judgement/requirement rather than a state of the world. However, since the action spaces of the norms aren't directly comparable, we need to decide how to combine action spaces. The easiest way is

to make the action space the product of the constituent action spaces (perhaps with the addition of the null action in each dimension), or simply to restrict actions to only change states in one role at a time. This certainly covers all possible social actions, but can certain action combinations be contradictory? This is what we touched on earlier when we said that one social norm can be an instance of another. This can be modeled if we show that the state space and action space of one model is an abstraction of another, and that the social Q-functions are somehow non-contradictory under this abstraction relation [perhaps the Q-value of an abstract (state, action) pair is the mean of the Q-values of the members of the extension of the pair. i.e. this is the measure of the correctness of an abstraction, so we can measure all possible abstractions - also the dynamics needs to be an abstraction]. Do such heirarchies help us make decisions by allowing us to decide on an abstract action first, or at least rule-out abstract actions?

8 Social predicates

The social state of a set of agents can be represented as a set of agent IDs (though, we'll see later that these may be inanimate objects or may not even exist) and a set of predicates on those IDs. A social predicate `pred(Alice,Bob)` is true iff Alice and Bob agree it is true. By agreeing to a predicate on oneself, one also agrees to take on the social role that goes with that predicate. Taking on a role is agreeing that one's acts will be socially judged by the social norms implied by the role.

Agent's actions can be split into two types: physical actions which change the physical state of subject and/or object, and social actions which change only the social predicates (abstracting over the physical brain state). To change the truth of `pred(Alice,Bob)`, Alice can suggest the change to Bob, if Bob agrees then the truth value changes by definition. The suggestion/agreement need not be verbal; if Alice lays out apples on a table with a price and stands behind it [we hardly notice here the social presumption of "behind" here rather than "next to" or even "in-front of, with back facing observer"], she's suggesting to anyone who passes that `marketTrader(Alice,PasserBy)`. If Bob stands on the other side of the table (not the same side as Alice, even though this would be physically more convenient) and inspects the apples, he is agreeing to the suggestion and so it becomes true. In this way we build a whole imaginary social world, and the social world becomes more important to us than the physical (but the physical acts as a substrate to the social).

By definition, any agent can unilaterally negate (but not assert) a predicate by simply withdrawing agreement (i.e. telling the other agent that it is no longer true), although that withdrawal may be judged to be socially unacceptable. However, if a social predicate depends on some phsical substrate for its truth, then an agent can forcibly negate the predicate by changing the physical substrate. If, in the above example, Mallory drives a car at high speed into the table of apples, then `marketTrader(Alice,Bob)` becomes false even though neither Alice nor Bob agreed. Mallory has performed a social as well as a physical action.

From these examples we see that there is a non-trivial relationship between physical actions and social actions. It's useful to distinguish between the objective physical world, which has its own dynamics without reference to social facts, and an agent's understanding of the physical world which, considering the importance of the social world, is likely to be instrumental in helping us understand the social world. It's also likely that we don't make a clear distinction between the physical and social worlds, the evidence being that we find it very hard to make any such distinction. The socio-physical world *is* our world. This can be modelled if we allow ourselves to be in predicate relations to inanimate objects. The semantics isn't very different if we consider a suggestion to an inanimate object consists of our attempt to bring about a phsical state in relation to the object and the objective physical world "decides" whether to "accept". This is even reflected in our langauge, for example "this nail is refusing to go in". The difference is that certain predicates can only be filled by other agents (though we seem quite fluid even on this).

8.1 Suggesting and telling

If Alice suggests a social tie to Bob and Bob agrees, then Alice believes the tie exists, as does Bob, but Alice also believes that Bob believes that the tie exists and Bob believes that Alice believes that the tie exists etc...so it is public knowledge between Alice and Bob. This doesn't require any truth-telling as the speech acts are social contracts, so agents are bound by the social tie.

If Alice tells Bob of some existing social tie, she is bound by the social contract that this is the truth, but Bob isn't bound to believe her. However, when society is functioning properly, the knowledge becomes public between Alice and Bob. It could be argued that the act of telling is an attempt to establish public knowledge, and if there is lying or disbelief then the attempt has failed.

So, the knowledge state of a set of agents can be reduced to the public knowledge between pairs of agents. This induces yet another graph where an edge denotes the existance of some public knowledge between two agents, and associated with each edge is a graph (or set of predicates) that is the content of that public knowledge. An agent's knowledge is then the sum of the public knowledge on its edges. However, if Carol witnesses Alice telling Bob some social fact, and Alice and Bob are aware of Carol's presence, then the fact becomes public knowledge between the three of them. This is distinct from three sets of pairwise public knowledge as it implies more (e.g. that Carol believes that Bob believes that Alice believes X).

8.2 What does it mean to be married to one's job?

8.3 Ownership

Ownership involves a predicate between an agent and an inanimate object, `owns(Alice,Cake)`, but constrains the behaviour of third parties. Alice can consider herself the owner of a cake without further ado, but this in itself constrains no behaviour. If Alice meets Bob she can tell Bob that `owns(Alice,Cake)` and

Bob can agree or dispute that. If Bob agrees, he is open to social judgement with respect to that ownership. Contrast this with the case when Alice suggests `pred(Alice,Bob)` to Bob. In a suggestion, Alice doesn't consider the suggestion to be true until Bob agrees, and Bob's agreement/rejection makes it true/false by definition. Whereas when Alice tells Bob something, she considers it to be true before the telling and irrespective of Bob's response.

Even if Bob agrees, he may still be physically able act on the cake and eat it. If he does, this will be judged as socially unacceptable, which will make it acceptable for Alice (or some representative of Alice) to punish Bob to the extent that justice has been served. Bob knows this and knows that this will likely happen so may decide not to eat the cake, even though he really likes cake.

Alice can give the cake to Bob by suggesting `-owns(Alice,Cake) & owns(Bob,Cake)` and Bob agreeing (here `-` means rescind, rather than not). The social norm would be to physically give the cake to Bob as well, although these concepts are distinct. On entering a room, objects held by (inside) an agent are assumed to be owned by that agent. If we assume agents can observe this, then this becomes a tacit understanding and agents are bound by this.

Alice, because she owns the cake, can give Bob permission to eat it without relinquishing ownership. This is different from giving him the cake as she retains the right to rescind the permission. Permission requires the reification of actions as welll as the new action of giving permission and perhaps a higher order predicate of having permission.

9 Social judgement and emotional reaction

A social action, or a failure to act, can be judged by others, and Alice's judgement of Bob will affect her behaviour towards Bob. This is public knowledge so Bob will consider this when deciding how to act. But what exactly does this public knowledge consist of?

Judgement does not specify exact consequences. If Bob eats Alice's cake, it may not be clear to anybody exactly what the consequences will be, but everybody knows it is not socially acceptable and may affect Alice's juedgement of Bob. Alice can also tell others that Bob ate her cake, affecting their judgement of him too. Bob can deny it but only at the risk of compounding his transgression by being found out and branded a lier.

There is a severity to judgement, but not an exact quantification. It is clear that eating Alice's cake isn't as bad as hitting her with a baseball bat, but there isn't a well defined calculus: is using her toothbrush better or worse than eating her cake? What if it was her wedding cake? So, Bob should expect her to react until she feels appeased. Nobody knows when this will be but everyone has a rough idea of the average. It is for others to judge whether she is overreacting.

Alice's social judgement of Bob, then, is best described as a change in Alice's emotional attitude towards Bob which affects her behaviour towards Bob. Her emotional attitude toward Bob changes in response to Bobs actions toward her and her actions toward Bob. So, if agents measure the quality of their

state in terms of other's emotional attitudes towards them, then at any instant they only need to choose the action that maximises some function of people's emotional attitudes and so the problem of living reduces to that of modelling other's emotional attitudes and seeing to ones bodily needs.

It is likely that we all share some innate emotional dynamic system that does not learn. Social norms, then, provide an interface between our socially constructed reality and our innate emotional system, allowing society to adapt through social change without having to evolve the emotional system through genetic change. The innate part can be modelled by giving each agent an emotional attitude towards each of its social ties. This attitude is changed by hard wired emotional reactions to certain changes to predicate states. Only a small vocabulary of "innate" predicates can be involved in the hard wired reactions. Social norms provide more complex social predicates and show how they are concretisations of the innate predicates. If agents are able to probe their own emotional reaction to an action, via it's effect on the innate predicates, then they are able to model others under the assumption that they have the same innate predicates and the same social norms. They are also able to model their own emotional state change in response to their social actions.

So, let an innate, *emotional system* be an action space, and an emotional state be a Poisson rate assigned to each innate action. There's also an innate input-action space, which describes actions others do to us. Each action (both input and output) is associated with a perturbation to the emotional state. Finally, there is an internal emotional dynamics which perturbs the state based only on the state (e.g. hunger and thirst intrinsically increase, perhaps anger exponentially decays). Each agent has an emotional state associated with each of their social ties (and perhaps a time-alone emotional state), and this is the mechanism that gives rise to the dynamics of the social role.

Emotional states are themselves pleasant or unpleasent (rewarding/unrewarding) to the subject. This gives an expected reward to playing a social role towards another agent, and so controls an agent's rate at which it will enter into social roles with another agent. This can be adaptive for different agents, so we learn to avoid/seek to play certain roles with certain agents (so an agent's model of other is, most simply, expected reward from playing each role to that agent). We also have a model of others and their social roles and emotional states which is used to interpret physical actions as abstract innate actions.

Undirected emotion: there are emotions that aren't directed at anyone. These control time-alone-behaviour (i.e behaviour that isn't interaction. e.g. if we allow hunger and thirst to be non-directed emotions, then I drink when I'm thirsty and eat when I'm hungry). Undirected emotion also plays the role of pure reward/punishment, there is no associated action but the emotion has an intrinsic reward. This can be a mechanism of sympathy/empathy as well as direct physical pain and can be elicited in other in order to reward/punish social action.

Innate actions live in a very abstract space, and will have a non-trivial relationship to physical actions.

[Can we also have directed emotion that isn't connected to action? I may care about you, but this may not be a particular action in a particular social role,

but will span across social roles and will affect the way I respond emotionally to your emotions. So, is caring about someone an emotion in itself, or is it a description of a state which reduces to propensities to abstract action?

Every agent has a model of the other agents with which it has social ties, within the model each agent is in a state which summarises what the subject agent has learned about his social ties from past experience. If an agent is unreliable, this will be recorded here. This encodes the agent's propensities to act within each social role and this learning will affect the way the subject agent will act towards its social ties. Perhaps this learning is at an emotional dynamics level: which implies we emotionally respond differently to one person compared to another even when playing the same social role. Perhaps I'm much more patient with you than someone else because I care more about you. Directed emotion that isn't tied directly to action could provide a rudimentary learning mechanism by changing how the other emotions translate to propensity to act: i.e. emotional state projects to act-propensity state and need not be the same dimension as the number of acts. - We can generalise this by saying there's a state space, an action space and a map from states to rate vectors on the action space, and input/output state change functions.

Alternatively, we can be more explicit about the distinction between directed emotions connected to actions and directed emotions that are parameters to the emotional dynamics. Given an approximation of the other's emotional dynamics, we can optimise our own emotional dynamics in order to maximise our reward.]

10 Innate Roles

Suppose that each agent has a set of innate social roles, whose state is expressed as abstract, innate predicates and whose actions are expressed as innate acts. The *e-function* is a map from state predicates to emotional state, which is a vector of reals, each of which is associated with one of the acts in the action space, and which gives the rate of activation of each act.

Note that the emotional state takes the place of the Q-function. Emotional state is a function of innate predicated state, this abstraction helps us deal with, and learn from, complex social situations.

So, it should be possible to describe every history as a history of concretisations of innate social roles. What is this innate vocabulary of roles/norms? [fairy tales? What happens if we put a lot of agents together who all have the fairy tales hard wired?]

Innate roles cannot be learned by individuals, but can evolve in a population of agents. Different innate roles will have different probabilities of reproduction.

11 Abstraction on a predicate space

Let a social state be a tuple (A, P_1, P_2), where A is a set of agents, P_1 is a vector of relations in $A \to \{0, 1\}$ (i.e. unary predicates over A) and P_2 is a vector of

relations in $A \times A \rightarrow \{0, 1\}$ (i.e. binary predicates over A). A social state can be made agent-centric by marking one of the agents as Self. So, given the sizes of A, P_1 and P_2 we can define a social state space. [In the graph representaion, does absence of a social tie imply that the predicate is false or that we don't know its truth value? In the case of ties that involve self, absence implies negation but in the case of relations between others, absence is lack of knowledge. Or, if we're claiming that a social graph encodes public knowledge among a population, then absence must imply negation in the sense that it isn't public knowledge, but this doesn't preclude the possibility that social ties exist that are public knowledge amond a different population. If an agent can belong to more than one population and a population can have public belief about agents outside of the population, then this can be used to represent asymmetrical knowledge].

Our aim is to define a tractable family of functions that assign emotional states to each ordered agent pair given a social state [However, emotional state depends on history of action: if Bob eats Alice's cake, she ends up without a cake, but it is Bob that she is angry with. So, either Bob becomes `stealer(Alice,Cake)` or the emotional change is a direct reaction to Bob's action, in which case we need a function that goes from changes to predicates to emotional reactions. The essence of Bob's eating of the cake is that he is forcibly removing Alice's `owns(Alice,Cake)` predicate. Alice must recognize this and reconize who did it. So, the action space becomes assert/negate/suggest each with a sentence (or at least a possibly negated predicate) and emotional reaction is to the act itself. This makes the function space much smaller.]

A simple mechanism to abstract over a social state is to remove some agents from A and remove all tuples that contain any removed agents from the extensions of the predicates. This makes most sense in agent-centric states, where we can, for example, remove agents that are unknown to Self.

We can also abstract over predicates by defining *abstract predicates* whose extensions are those of sentences over the predicates of the state space, where the arguments of the predicates in the sentence refer to the arguments of the predicate (e.g. $IsAFruit(X) \equiv IsABanana(X) \vee IsAnApple(X) \vee ...$). This generates an abstract social state space with the same set of agents but a different (potentially smaller) set of predicates. This abstract space can itself be abstracted over in the same way, or by agent removal. This may require an extension to predicates with more than two arguments, but since we're defining their extension as a sentence rather than enumerating tuples, this poses no problem. This mechanism allows us to build a type heirarchy and typed predicates, among other things. In the context of roles, these are abstractions over the state spaces of some number of roles. We can reduce the number of states in a single role, or unify multiple roles into a single role.

A function from an agent-centric social state to the reals can be described as an equation expressed as functions and sentences on the predicates with existential quantification over all agents except Self. Where there is a sentence, its value is 1.0 if true and 0.0 if false. I think this reduces to a weighted sum of unary predicates (which are equavalent to existentially quantified sentences) which take a Self. An emotion vector can then be described as a vector of weighted sums of unary predicates which, when instantiated with an agent, gives the emotional state of that agent.

Connecting this to our discussion of innate predicates, the emotional state of an agent is a weighted sum over (existentially quantified sentences over) the innate predicates, with special reference to Self. Social norms form a concretization of the innate predicates (so in this picture, evolution goes from abstract innate to more concrete social).

Every agent has an agent-centric representation of the social state of their world. When society is functioning correctly, everyone agrees which role each agent is playing towards other and what state in that role they are in. This is represented by binary predicates, so for any two agents, there exists a mapping from the agents in one anget's state to that in the other's which includes, at least, each agent's Selves, such that the union of their states is consistent w.r.t. the binary predicates. [does binary consistency imply global consistency?] Consistency requires that predicates that represent states in the same role are mutually exclusive [perhaps we could represent this as].)

12 Abstraction of actions

The physical environment defines which social actions are available to an agent at any time. We suppose there is a dynamic system that includes all the objects and agents in the world including the brains of the sentient agents. Our agent based model should be an abstraction of this dynamic system. In our model, each agent has an agent-centric model of others, and this should be an abstraction of the physical state of the agent.

At base level, we assume agent's observations of the world around them cause changes in their physical predicates, and actions change subsequent observations. From this we can model an agent's physical state as the possible observations that may be made. If we assume an agent is fully aware of its physical state then we can take the physical state of an agent to be its physical predicates and allow other agents to observe some or all of these. So, we define a set of physical predicates (perhaps different for each type of object) and a set of actions an agent can perform on an agent, given its physical state, and the state change resulting from those actions. This should be a closed system with no reference to any social or innate state.

However, the observable physical state does not include the brain state of the agents. In consequence, the social state is not an abstraction of the observable state, it includes information not contained in the physical state. However, a physical action must be interpretable as a social action given the current observable and social state, and if an agent intends to perform a social action they must be able to generate a set of physical actions that will achieve this.

12.1 '80s adventure game physical worlds

We use as a physical world a multi-agent version of the physical worlds of '80s adventure games where agents can move around a set of locations and interact in simple ways with various obejcts they find at those locations.

Let a *physical world* be a tuple (A, P, α) where A is a set of agents and objects

in the world, P is a set of unary or binary predicates, α is a set of actions which are $(oVars, iVars, p, c)$ tuples where $oVars$ are open variables which must be filled to instantiate the action, $iVars$ are a set of variables with unique ("the") quantification, p is a sentence giving the prerequisites that are required to be true for the action to be possible and c are the consequences of performing an action in terms of changes to the state of the world. We can write this as

$$X : \iota I : P(X, I) \underset{a}{\rightarrow} C(X, I)$$

which is to be understood as distinct from implication.

The state of the world is a set of true predicates in P applied to objects in A. In the case of an objective state, the absence of a predicate implies its negation. In the case of an agent-centric state, the absence of a predicate involving Self implies its negation, but the absence of any other predicate makes no commitment to its truth. In the agent-centric representation, the subject of all actions is implicitly assumed to be Self, so can be hard-wired into the prerequisites and consequences. In the objective representation, the Self is taken to be an open variable.

12.2 Containment

All objects can be arranged into a tree structure where the root of the tree is the `world` and every child is related to its parent with the `isIn(Child,Parent)` relation, which is to be understood that the child is contained within the parent.

Only siblings and (parent,child) pairs can interact. Agents can `pickup` their siblings and `putdown` their children.

For example `pickup(S,O)` can be written

$$\begin{aligned} S, O &: \iota P \\ &: \texttt{isIn(O,P)} \wedge \texttt{isIn(S,P)} \wedge \texttt{canContain(S,O)} \\ &\underset{pickup}{\rightarrow} \texttt{isIn(O,S)} \wedge \neg\texttt{isIn(O,P)} \end{aligned} \tag{1}$$

[what about agency here? How do we distinguish S picks up O from O crawls into S and O falls into S from B puts O into S? So, an agent's action changes the state of the world given some prerequisites. The world itself doesn't care about agency, just state, but social knowledge records the agent of an action. Actions consist of an actor and a set of updates to the predicates (possibly introducing new objects). A partial time ordering of actions is maintained by specifying that the action took place after the latest actions involving each of the objects named in the modification.]

When an agent performs an action and updates some predicates, other agents observe this change, but part of the observation is the identification of the subject that acted to make those changes. Or, alternatively, the action itself implies a social interpretation [how?] and so changes the actors social state and possibly other agents' social state.

12.3 Locations

We can create a mapped world in which the agents can roam by adding objects that are located relative to eachother via spatial relations such as `isNorthOf` and `isEastOf`. Each location can contain a number of objects, including agents, by asserting `isIn(Obj,Loc)`. An agent can move between neighbouring locations by performing the `moveNorth`, `moveSouth`, `moveEast` or `moveWest` actions.

For example `moveNorth(S)` can be written as

$$S : \iota A, B : \texttt{isIn(S,A)} \wedge \texttt{isNorthOf(B,A)} \underset{moveNorth}{\rightarrow} \texttt{isIn(S,B)} \wedge \neg\texttt{isIn(S,A)}$$

In this way we can build up a purely physical world that has no reference to the social world.

Locations are effectively domains of observation and action. An agent is always in a location and can only ineract-with/observe objects/agents in the same location. Objects in other locations must be remembered. We can assume that all objects in a location form a public knowledge group on both physical and social predicates.

12.4 Exchange as social norm

[How do we explain exhange in terms of social norms?]

[Exchange already seems to require a concept of justice, or at least some way to negotiate the exchange...although we've already shown that pure Q-learners can learn to exchange].

12.5 Belief, public knowledge and social facts

We can extend a physical world domain to the socio-physical world by allowing predicates that, on a physical level, refer to the states of agents' brains. These will specify the attitudes each agent has towards other agents, which will affect the way an agent behaves towards the other, and an agent's expectations about another's behaviour towards them. Higher order expectations (or beliefs about expectations) are possible but here we assume only first order expectations. For example `customerRole(S,O)` means that agent `S` is bound by the role of customer towards agent `O` (and at minimum, `S` and `O` know this). We call these *social predicates*, they can be unserstood as meaning that an agent has (or ought to have) some attitude towards another agent or object. If an agent has an attitude, we call that a *social fact*.

To reason about reasoning agents we need to include a `believes(S,`Θ`)` predicate meaning agent `S` believes Θ. However, social relations avoid this by relying on *public knowledge*. A fact, f, is public knowledge among a group of agents if every agent knows f and knows that f is public knowledge. So, if Alice and Bob are in the group, this implies that Alice knows Bob knows f and Bob knows that Alice knows he knows f etc...

In the simplest model, agent pairs exchange promises to be bound by roles with respect to eachother, but have no social knowledge beyond that. This model

induces a directed graph of promises. We assume each agent knows every true physical predicate, the promises it has made and the promises made to it.

An alternative simple model would be to assume that all social and physical facts are always public knowledge among all agents (this is approximated by gossip). In this model, all agents have the same belief state so we need only represent this once.

In a more complex model, agents can only observe the physical and behavioural predicates of the location they are currently in, and each forms a view of their world as a database of predicates from these observations.

Agents can individually or collectively have attitudes towards objects that do not have any physical existance (e.g. Microsoft, God) and so we need to extend the domain of obects to include these. [although we need to explain how two agents come to refer to the same non-physical object and the types of relationship between non-physical and physical objects. Note that it is possible for a belief about Microsoft to be falsified.]

If we model an agent's beliefs about the physical world, an agent can (mistakenly) have beliefs about an object that doesn't physically exist. An agent can then act on the non-existant object. At this point, the physical world makes the agent aware that the object does not physically exist.