Social Software

Rohit Parikh
City University of New York
Departments of Computer Science, Mathematics and Philosophy
CUNY Graduate Center
365 Fifth Avenue
New York, NY 10016[1]

Abstract

We suggest that the issue of constructing and verifying social procedures, which we suggestively call social software*, be pursued as systematically as computer software is pursued by computer scientists. Certain complications do arise with social software which do not arise with computer software, but the similarities are nonetheless strong, and tools already exist which would enable us to start work on this important project. We give a variety of suggestive examples and indicate some theoretical work which already exists.*

> I send someone shopping. I give him a slip marked "five red apples". He takes the slip to the shopkeeper who opens a drawer marked "apples"; then he looks up the word "red" in a table and finds a colour sample opposite it; then he says the series of cardinal numbers – I assume he knows them by heart – up to the word "five" and for each number he takes an apple of the same colour as the sample out of the drawer.
>
> "But what is the "meaning of the word 'five'?" No such thing was in question here, only how the word "five" is used.

In this passage from the *Philosophical Investigations* Wittgenstein is describing a social algorithm, albeit a simple one. He also introduces the notion of a *data type* (though not by that name) which is now quite important in computer algorithms. The words "apple", "red", "five" belong to different data types and are used in very different ways. This variety forms a sharp contrast to the uniformity of objects in set theory, where everything, natural numbers, reals, etc. are constructed from the same basic material. But it does form a parallel to the variety we find in computer algorithms. In computer algorithms we also find *integers*, *stacks*, *queues*, and *pointers* which play different sorts of roles.

Wittgenstein's purpose in his example is to wean us away from the notion that there is just one kind of thing – meaning – which explains all different kinds of words, and he uses

[1]email: ripbc@cunyvm.cuny.edu Short ascii messages only; no attachments can be read by this account. The author is also affiliated with the Department of Computer Science, Brooklyn College of CUNY. Research supported by grants from NSF and the PSC-CUNY FRAP program.

his language games as a way of re-iterating the primacy of use as well as the variety of ways in which words are used.

But something important in human practices is left out by Wittgenstein, namely *why* do these people act as they do? Perhaps we can put aside that question with the shopkeeper as we are ourselves used to shopping, but the issue arises again with Wittgenstein's example of the builder and his assistant. Wittgenstein's builder uses various words like "block", "pillar", etc., and his assistant brings the corresponding objects. *Why* the assistant does this is not a question which is raised. Perhaps the assistant is paid, perhaps he is under some threat, or perhaps he worships the builder. It does not matter for Wittgenstein. But of course for any genuine theory of social software it *does* matter. There is not much point in designing a social algorithm for some purpose unless one has provided for people to *want* to carry out that algorithm.

Perhaps the point for Wittgenstein is that utility or purpose need not matter for an explanation of meaning. But the issue is important for Grice. Grice wants to explain why logical symbols like "or" or "implies" as used in natural language seem to have different properties from their formal counterparts $\vee$ and $\rightarrow$. In explaining this difference, Grice introduces the notion of *implicature*, a notion which is primarily pragmatic.

For instance, If my car is out of gas and you say that there is a gasoline station around the corner, you have implicated, but not said, that the gasoline station is open. If you say that someone for whom I am looking is in the south of France, you have not said, but implicated that you do not know more precisely. The implicature comes from the fact that you have some idea of my needs in asking some question and your answer is expected to be as helpful as it can be, given your state of information.[2]

Grice's point is that language is a human activity where people help each other to perform tasks. In a conversation, the help takes the form of relevant information, and the consequent requirements of relevance and helpfulness go beyond the requirement of simple truth. In as much as language is performing such a utilitarian role, Grice's maxims like, "Avoid obscurity", "Avoid ambiguity", etc., make sense. Grice sees the parties in a conversation to be taking part in a game of co-operation, and explains his maxims and their linguistic impact as coming out of the need to co-operate.

The co-operation can take odd forms at times. A butler entering a hotel room to clean it encounters a woman guest coming out of the shower. "Excuse me, sir", he says and withdraws. Here he is implicating that he did not see the guest clearly, thereby saving her embarrassment. But this kind of implicature is different from the usual one, as there is no common knowledge here, e.g. of the fact that the guest is female, nor is there common belief that the guest is male. What we have instead is *helpful deception*.

[2]One difference between the implicature of an uttered sentence and its meaning is that the implicature is normally cancellable. For instance you could perfectly well say, in the first case, "There is a gasoline station around the corner, but I do not know if it is open", or in the second, "He is in the south of France, and actually I know more precisely, but am not at liberty to tell anyone".

Grice has brought in the notion of utility which was absent from Wittgenstein's examples, but of course he does not study utilities formally as game theorists do. Grice's purpose is to sketch a rough outline of a theory and leave the details to the future. Moreover, Grice is not concerned with buying and selling, activities both like and unlike conversation, which game theorists are interested in.

Since my purpose here is to study social software in general, I shall not confine myself to language, nor to buying and selling, but will look at the nature of social software in very general terms. The first task is to raise the twin questions of why such a study is important and why it is feasible.

It is quite usual these days for an off the shelf PC to be able to carry out tens of millions of multiplications every second. By contrast, it took New York City around ten years to contemplate the Westway project, study it, and finally not carry it out. There is a striking contrast between the efficiency with which purely computational processes are carried out, and the inefficiency of the social processes which are intended to be, and sometimes actually are, to the benefit of all of us.

There are many reasons why social procedures do not always succeed or when they do, are cumbersome and give rise to unintended ill effects. This fact may seem to be part of the domain. People just aren't as tidy and well behaved as computers, they are willful and forgetful and selfish. Moreover, different people have different ideas of what is the best thing to do in any given situation so that conflicts can arise even between well meaning individuals.

Given these difficulties it may seem that we should accept whatever we can get in terms of the relative success of the social procedures we do have.[3]

I want to argue that even though all the difficulties mentioned are real and no doubt we shall never have social procedures which work ideally, we *can* nonetheless have a theory of social procedures which is analogous to the formal theories for computer algorithms which exist in computer science. I am referring here to a whole group of theories, some of which have come into existence during the early seventies and some are newer.

These are
a) the theories of *program correctness* which seek to prove that computer programs achieve the purpose (called the specification) which they are intended to achieve.
b) the *analysis of programs* which seeks to analyse the efficiency of programs in terms of resources utilized and the amount of time taken and
c) *Concurrency theory*, and *Distributed computing* which analyze the behaviour of several computing processes acting together and which must ensure that different processes

[3] I shall use the terms algorithm, game, procedure, more or less equivalently.

sharing some resources do not frustrate each others' purposes and do share information so that when a process needs to act, it knows the facts that it needs to decide which action to take.

Can there be analogous theories in the social domain – theories which tell us how to construct a social procedure which brings about a desired result in an efficient and reliable way, at least when such a thing is possible at all? I shall give some examples and partial results which suggest that such a thing may be possible; and surely no one would deny that were it possible, it would have tremendous potential for good. It might even have theoretical interest if we are able to formulate something analogous to the Church-Turing thesis as to what is socially achievable and what is not.

It so happens that we do have an analogue to one of these three concerns, namely Game theory is an analogue both to concurrency theory when we consider agents acting concurrently and in ignorance of each others' moves, and to distributed computing, since we also consider situations where the agents are acting in turn, in full knowledge of each others' moves.

Surely it is because of these similarities that the issue of knowledge is important in both distributed computing and in game theory, and the communities in both areas have participated in the *TARK* conferences which began in 1986 and have continued to this day.

In proving correctness, one approach has been Game logic. Early work on Game logic in the early 80's [Pa83,Pa85] showed how the correctness of the (Banach-Knaster last dimisher) cake cutting algorithm could be proved formally. Here the correctness consists of showing that each player has a winning strategy and can get a fair portion of the cake regardless of the actions of the other players. This work has been followed up recently, especially by Marc Pauly, who has constructed a formal logic to reason with coalitions.

But there is not yet a *general* formal theory to prove to us that social procedures achieve the desired effect. Since the structure of games in terms of simpler subgames is not analyzed very much in Economics, there is no genuine analogue to the analysis of algorithms. Games like the centipede game do have a structure, but it is rather simple and there is no general theory of how complex games can be constructed from simpler ones. There are of course certain special cases. For instance the *Prisoners' Dilemma* which is problematic is often replaced by the *repeated* prisoners' dilemma. Thus we do have *iteration*, a common construct in program development. But other constructs, and even a systematic study of iteration are lacking.

Before we try to extend ordinary algorithmic intuitions to the social setting we need to recall that unlike computers, people do not obey orders, or at least not reliably. So in order for a complex procedure to be carried out, people have to be given inducements to take part in such a procedure. Game theory, whatever its limitations, allows us to formalize the notion of the incentive to do something rather than something else. Roughly speaking, if an agent has a choice among several actions, we would expect that the agent will carry out that (pure) action which the agent thinks will bring the maximal benefit (utility) to the agent. Since the benefit may depend on the actions of other agents, the agent also may need to have some theory of *their* probable actions which may be backed up by some theory of what they want and what they think about the probable actions of yet other agents.[4] It is quite likely that crowd actions like a riot or a round of applause happen when the beliefs of the agents about the actions of other agents suddenly change. If I think that a sonata has ended but am not sure, I will hesitate to applaud unless I see and hear others clapping. When they do clap, I am safe from ridicule and I will join in. Such sudden movements where perception changes arise also with stocks, and when large enough, a crash in the stock market can occur.

Thus a successful social algorithm will depend on creating incentives for all agents to act in a certain way and also on creating certain perceptions which back up the actions of agents by making them think that the other agents will act in a suitable way. For example if I am a stand up comic and I want the audience to laugh at certain points, I may plant some friends in the audience whose function it is to laugh on cue. Once my friends laugh, others in the audience will feel that the joke is a safe one and not in such bad taste that no decent person would laugh. The facts that my friends are laughing will then cause general laughter, with many or most joining in.

In (Chwe 2001) Michael Chwe discusses the role which common knowledge plays in cultural rituals giving a variety of rich examples. Examples include patterns in advertising, the role played by royal processions, and the shape of the theatre in which plays were staged in Greece (conducive to common knowledge). Common knowledge, or something close to it, can be decisive because an agent who feels that there is safety in numbers may act in concert in ways in which the agent may be reluctant to act alone.

In the movie *The Messenger*, after Joan of Arc first arrives to meet the Dauphin, she is introduced to an archer who will find out what her needs are. They are, a sword, a war horse, a suit of armor, and a banner. The function of the first three is clear. But what about the *banner?* Its function is largely informational. During a battle, the banner creates common knowledge among the soldiers of the fact that Joan of Arc is alive, and moreover, where she is. The first creates in them the resolve to fight on, and the second serves to co-ordinate their actions.[5]

To sharpen our intuitions, I shall start with some more specific examples. Unlike the examples used in game theory textbooks, most of these will be real life or close to real life examples to indicate that the theory has potential applications.

One way functions: let me illustrate with an example. Suppose I give you the name of someone who is listed in the telephone book and ask for her phone number. Then you can find it easily by just looking up the name. Suppose, however that I give you someone's number and ask you for the name. You can also find *that*, but essentially only by looking

[4] See the paper by Brandenburger and Keisler for a possible paradox which can arise here.

[5] The ubiquitousness of the US flag in the aftermath of the WTC bombing seems to have a purely emotional role, without the informational aspects of Joan's banner.

through the entire directory. There is no easy procedure. So we have a function f from names to numbers which is easy to 'compute' and whose inverse g from numbers to names, while also computable, is cumbersome and time consuming. The telephone directory is a one-way function.

Now for our actual example:

At the *American Philosophical Association* (APA) meeting in New York (Dec 2000), we had an example of a one way function at work, in the *wrong direction*. The program booklet gave, for each talk and session the *name* of the room in which the talk or session was taking place. Thus for example many logic sessions were in Gramercy. The program booklet also gave a *floor map* for each of the four floors of the Hilton hotel where the conference was taking place. Thus given the program and the map, one could find out in theory where a given talk was. However, the maps constituted a one-way function in the wrong direction. What they had was the sort of information, 'what is the name of the room at the north east corner of the third floor?', and not 'What is the location of the room with a given name?' In other words, given a location, the maps gave you the name of the room at that location. The program itself, on the other hand, gave you the *name* of the room where a particular talk was taking place. Thus we had two functions, the booklet gave you a function f : *events* $\longrightarrow$ *room names*. The maps gave you a function g : *locations* $\longrightarrow$ *room names*. Clearly one could not compose them without inverting the function g. Hence there was no easy way to find the location from the name of the room. You could find the location using the maps, but only by looking through all the floor maps.

It would have been easy to construct the program booklet in such a way that one could use it to go directly to the room where a talk was being held, without going through four maps. Instead of saying only that the Logic session was in Gramercy suite, the booklet could have told you that it was in Gramercy, on the south side of the second floor. A great deal of time could have been saved for people running around, looking in vain for the room of their session and many of whom missed the first few minutes of the talk. What the *APA* booklet did was like giving you a phone book which allowed you to find names from numbers, when what you needed was one which gave you numbers from names.[6]

This example has some interest since there are no game theoretic issues here, no one wants someone else to go to the wrong room for some talk (I hope). It is merely an issue of algorithmic efficiency.

The Carousel example: Many of us prefer to carry our luggage on board when taking a flight. However, sometimes we *have*, for various reasons, to check our baggage. Then on arrival one has to wait at a moving carousel which brings all pieces of luggage one after another. If we are lucky, ours is among these. However, a curious phenomenon takes

[6]Some maps do indeed give you a key whose effect is to diminish the wrong way function problem. Thus a street map may be divided into squares and there may be a key which, given the street name, gives you the square in which the street lies. But these keys still tend to leave a lot of searching to be done.

place which resembles the problems of *the Prisoners' Dilemma*, or *The Tragedy of the Commons*.[7] One gets a better view of the approaching suitcases if one goes closer to the carousel. But by doing this, one inevitably blocks the view of one's neighbour who then also proceeds forward towards the carousel. When this process is finished, all passengers are right at the carousel, blocking each other's view, and every one is worse off than if no one had walked up to the carousel.

However, there is a difference between the carousel problem and the other two problems, which makes the carousel problem a social programming problem. In the classic prisoner's dilemma for instance, the jailer hopes to use the game theoretic pressure on both prisoners so as to get them to confess, send them to prison, and win a promotion. With the carousel, the airline has no such malicious motive. Indeed their profits will be greater if the passengers find their flight experience less unpleasant, and are motivated to use this airline again.

And the airline *does* have a solution to the carousel problem. All they need to do is to paint a line about 18 inches from the carousel and post signs saying 'Do not cross the line until you actually see your own suitcase approaching'. Then social pressure would keep people from crossing the line unnecessarily, and everyone would enjoy a better view of the oncoming luggage. Subways routinely do something similar at platforms to prevent passengers from falling onto the tracks or being hit by an incoming train. There is an orange line about 18 inches from the platform edge which you are asked not to cross until the train is in the station and has come to a complete stop. Surely many accidents are avoided this way, and the same method would work for baggage carousels. But somehow, since the carousel problem is not life threatening, no one has thought of this simple solution.

In Chopra and Parikh (1999) the problem of overbooking by airlines is considered. Airlines typically overbook so that the set of promises made by the airline is inconsistent with the physical fact of the plane's capacity. An inconsistency tolerant logic is then needed to handle the airline's verbal behaviour (which may be described by its customers as lies!). However, algorithmically, the airline can be successful if the number of passengers actually claiming seats is less that the capacity of the plane or exceeds it only slightly.

Resource sharing: The Sante Fe bar problem, first discussed by Brian Arthur (1994) (see also Greenwald, Mishra and Parikh 1998) goes as follows. There is a certain bar in Santa Fe where Irish music is offered on some nights. There are (say) a hundred people who would like to go, but the bar has room only for 60. People prefer to go if the bar is not crowded, but prefer *not* to go if it is. Now suppose that people have (the same) data on previous weeks, how many people went etc. And they want some theory which utilizes this data to tell them whether the bar will be crowded *this* week. There cannot be a correct theory because if there were, surely they ought to believe it, but then if the

[7]The tragedy of the commons arises when several families from a village share the same pasture for grazing their cows. Since no one has an incentive to limit the size of his herd, overgrazing takes place and the grass dies.

theory said that the bar would be crowded, they would all not go and the bar would not be crowded. If the theory said that the bar would not be crowded, they would all go and the bar would be crowded. This is reminiscent of the Russell set R which has the property that $R \in R$ iff $R \notin R$.

Greenwald *et al* discuss several kinds of equilibria and learning bahaviours which can arise when prospective bar-goers try to balance their desire to go, with their information about the attendance in previous weeks, together with the knowledge that the bar could very well be crowded on just the day they choose to go. Ironically, the desire of the bar-goers to maximize their own pleasure conflicts with the most efficient use of the bar itself and one device suggested by Greenwald *et al* is a system of taxes imposed on the bar-goers, the proceeds of which are shared among those who do not go. A simulation shows that such a system works against (destructive) individual selfishnes and maximizes total (social) satisfaction.

Such problems arise in other contexts when the demand exceeds the supply. The free market solution is to let prices rise to diminish demand. For instance the bar could raise its prices high enough to drive out 40 potential customers. But many consider this sort of solution to be unfair. Another solution is rationing, but that involves some *big brother* making decisions and is too inflexible. What if Jack wants more than his share of coffee but is willing to take less than his fair share of sugar, whereas with Jill it is the other way around? Rationing may leave them both unsatisfied unless Jack and Jill happen to run into each other.

Another similar puzzle arises when a highway which can handle 10,000 cars per hour is shut down for roadwork. There are two older roads, A and B, each of which can handle 5,500 cars. If we put a *detour* sign at the entrance to the highway, what should it say? If it says to take the road A, it would be jammed. If it says to take the road B, that would also be jammed! Could we have a sign saying, "Take route A if you are a Democrat and route B if you are a Republican"? [8] Resource allocation and sharing require information transfer and some sort of utility pressure and creative solutions are needed to bring that about.

For a practical case of this, after the World Trade Center disaster in New York City there were calls for people to give blood. However, many more people responded than were needed and there were long lines of donors at the hospitals. If ten thousand pints of blood are needed and an appeal is made to a socially conscious city with a population of eight million, then such a problem is bound to arise. But if no appeal is made then of course there *would* be a shortage of blood. So again the same sort of problem arises as we saw in the Santa Fe bar problem. A person who goes to give blood when her blood is needed is very happy to help whereas a person who goes to find a surfeit of donors is going to be frustrated.

[8] Marc Pauly points out that this could lead to accidents involving Buchanan and Nader's third party followers!

The two horsemen: Suppose we want to find out which of two horses is faster. This is easy, we race them against each other. The horse which reaches the goal first is the faster horse. And surely this method should also tell us which horse is *slower*, it is the other one. However, there is a complication which will be instructive.

Two horsemen are on a forest path chatting about something. A passerby M, the mischief maker, comes along and having plenty of time and a desire for amusement, suggests that they race against each other to a tree a short distance away and he will give a prize of $100. However, there is an interesting twist. He will give the $100 to the owner of the *slower* horse. Let us call the two horsemen Bill and Joe. Joe's horse can go at 35 miles per hour, whereas Bill's horse can only go 30 miles per hour. Since Bill has the slower horse, he should get the $100.

The two horsemen start, but soon realize that there is a problem. Each one is trying to go slower than the other and it is obvious that the race is not going to finish. There is a broad smile on the canny passerby's face as he sees that he is having some amusement at no cost. Figure I, below, explains the difficulty. Here Bill is the row player and Joe is the column player. Each horseman can make his horse go at any speed upto its maximum. But he has no reason to use the maximum. And in figure I, the left columns are dominant (yield a better payoff) for Joe and the top rows are dominant for Bill. Thus they end up in the top left hand corner, with both horses going at 0 miles per hour.

	0	10	20	30	35
0	0, 0	100, 0	100, 0	100, 0	100, 0
10	0, 100	0, 0	100, 0	100, 0	100, 0
20	0, 100	0, 100	0, 0	100, 0	100, 0
30	0, 100	0, 100	0, 100	0, 0	100, 0

Figure I

However, along comes another passerby, let us call her S, the problem solver, and the situation is explained to her. She turns out to have a clever solution. She advises the two men to switch horses. Now each man has an incentive to go fast, because by making his competitor's horse go faster, he is helping his own horse to win! Figure II shows how the dominant strategies have changed. Now Joe (playing row) is better off to the bottom, and Bill playing column is better off to the right – they are both urging the horse they are riding (their opponents' horse) as fast as the horse can go. Thus they end up in the bottom right corner of figure II. Joe's horse, ridden by Bill comes first and Bill gets the $100 as he should.

	0	10	20	30	35
0	0, 0	0, 100	0, 100	0, 100	0, 100
10	100, 0	0, 0	0, 100	0, 100	0, 100
20	100, 0	100, 0	0, 0	0, 100	0, 100
30	100, 0	100, 0	100, 0	0, 0	0, 100

Figure II

Of course, if the first passerby had really *only* wanted to reward the slower horse (or its owner) he could have done this without the horses being switched and for a little extra money. He could have kept quiet about the $100 and offered a prize of $10 to the owner of the faster horse. Then when the race was over, he would hand over the $10 to Joe and $100 to Bill. Here the effect would be achieved by hiding from the two horsemen what their best strategy was, and to fool them into thinking that some other action was in fact better.

While the problem of finding the faster horse, and that of finding the slower, are equivalent algorithmically, they are not equivalent game theoretically when the men ride their own horses. The equivalence is restored when the two men switch horses.

For a practical analogue of the two horses example, consider the issue of grades and letters of recommendation. Suppose that Prof. Meyer is writing a letter of recommendation for his student Maria and Prof. Shankar is writing one for his student Peter. Both believe that their respective students are good, but only good. Not very good, not excellent, just good. Both also know that only one student can get the job or scholarship. Under this circumstance, it is clear that both of the advisers are best off writing letters saying that their respective student is excellent. This is strategic behaviour in a domain familiar to all of us. Sometimes employers will try to counter this by appealing to third parties for an evaluation, but the close knowledge that the two advisers have of their advisees cannot be discovered very easily. And unfortunately, there is no obvious analogue to the strategem of exchanging horses.

Shankar's choices

Meyer's choices

	G	VG	E
G	*NJ , NJ*	*NJ , J*	*NJ , J*
VG	*J , NJ*	*NJ , NJ*	*NJ , J*
E	*J , NJ*	*J , NJ*	*NJ , NJ*

Figure III

In Figure III above, *J* represents job and *NJ* represents no job for the student. Then Meyer's lower strategies dominate his upper ones. And for Shankar, his rightward strategies dominate the strategies to the left. Hence, with each playing his dominant strategies, they end up in the lower right hand corner with neither student getting the job.

Bills in restaurants: When we go to a restaurant the waiter brings a menu. We order, then the food comes and after we eat, we pay. All of this is a social ritual which ends with the waiter computing a charge and with us paying for the total cost including tax and tip.

But what is the function of this ritual of paying? Or what is the *specification* which this algorithm fulfills? Clearly the function is to make sure that we get the food that we want, and that the restaurant manages to make enough money to stay in business. These are the two requirements. The 'paying for what you ate' ethic is normally used but is not essential.

For example, during a lunch buffet, Joe eats 2 pounds of food for which he pays $13.95 plus tax and tip. Bill and Sally together eat 1.8 pounds of food for which they pay $27.90 plus tax and tip. They ate less, but they pay more. If Joe had brought his friend Edna along and they had shared the 2 pounds of food, the restaurant would not have accepted a payment of $13.95, which is only for one person. Thus the buffet restaurant is not charging by the quantity of food consumed but by the number of people eating. The two methods are not equivalent in detail. A big eater will benefit from a buffet and a frugal eater will benefit less or will lose out. But as long as the prices are reasonable, both may be willing to eat the buffet, and the restaurant will make enough money to stay in business.[9]

Dim Sum restaurants often use quite another method. The customers select what they want out of carts which constantly go by. There is no formal menu as such, but dishes with different prices are kept in plates of different colors. Thus all blue plates might have contained dishes costing $4 and all green ones might have contained dishes costing $3. At the end of the meal, if there are six blue and eight green plates left on the table, then the cost of the meal is $24 + $24 or $48.

Thus the same specifications, satisfied customers, solvent restaurant, can be fulfilled by different procedures. Once we know what we want in a certain situation, we can then proceed to try to work out a procedure which fulfills our desires.

Vagueness and social algorithms: in the *Philosophical Investigations*, (paragraph 88) Wittgenstein asks:

> If I tell someone "Stand roughly there" – may not this explanation work perfectly? And cannot every other one fail too?
>
> But isn't it an inexact explanation? – Yes; why shouldn't we call it "inexact"? Only let us understand what "inexact" means. For it does not mean "unusable".

I would like to suggest that a degree of exactness is sufficient if it is adequate for the particular social algorithm. For example suppose a group of people going for a picnic are supposed to meet at Grand Central at 9 AM. It may not matter if one of them is a minute late. But if they are planning to take the 9:10 and that person is 11 minutes late, they will miss the train and the algorithm will break down. If that person is only 5 minutes late, they will make the train, but only through someone else having to suffer the aggravation of buying the late person's ticket for him. It is clear that there is no sharp distinction here of success vs. failure of the algorithm. Perhaps even if the 9:10 is missed, there is the 9:40. But the picnic will be shorter.

No feasible algorithm can require *perfect* exactness. But different algorithms may require different amounts of exactness, and when one says, "Be here at 9 AM sharp!", one is not asking for a degree of exactness which is perfect, but for a degree which is more than customary.

Vague predicates also suffer from the Sorites kind of difficulties, first mentioned by Eubulides. They do not have precise extensions, and cases where a predicate clearly applies can

[9] Wittgenstein makes a similar point. "These people – we should say – sell timber by cubic measure – but are they right in doing so? Wouldn't it be more correct to sell it by weight – or by the time that it took to fell the timber – or by the labour of felling measured by the age and strength of the woodsman?" *RFM* I-148

shade gradually into cases where it clearly does not. Social algorithms have to provide for these difficulties too.

For instance, the color green, which indicates *Go* in traffic signals can gradually shade into yellow, which means *Slow*, which can gradually shade into red, which means *Stop.* If traffic signals were allowed to be of various colors, then different observers might disagree on whether a particular signal was yellow or green. This could lead to accidents. To avoid this problem we use signals of three fixed colors only. The space of all colors which is connected and continuous, allows for Sorites type paradoxes. But we replace it by the space of only three colors, which is immune to such problems (Parikh 94, PPP 2001).

Solomon and Gibbard-Satterthwaite: In his survey paper John Moore (1992) discusses Solomon's dilemma (*The Old Testament, the first book of kings*) and Solomon's game theoretic solution. Two women, say Anna and Bess, claim the same child and come before Solomon to adjudicate. Solomon threatens to cut the child into two and share it between the two women, at which the real mother protests and expresses willingness to give the child to the other. Once Solomon realizes who the real mother is, his problem is solved.

Let us look at this game theoretically. There are two possible states of nature, α in which Anna is the real mother and β in which Bess is the real mother. Solomon does not know which is the actual state, but the two women do. Moreover, if we say that there are three possible decisions (Moore considers four), a for giving the child to Anna, b for giving the child to Bess and k for killing the child, then the preferences go as follows:

In state α Anna's preferences are $a > b > k$. Anna would like her DNA passed on (not that Solomon thought of this) so that she would prefer the child to be brought up by Bess than to have it killed. Bess on the other hand merely wishes to experience the pleasures of parenthood. She dislikes Anna so that she prefers to have the child killed than to have Anna enjoy it. Thus Bess' preferences are $b > k > a$.

In state β, it is the other way around. Anna's preferences are $a > k > b$ and Bess' preferences are $b > a > k$. Thus regardless of the actual state of nature, the real mother will prefer to give up the child than have it killed whereas the other woman's preference will be to have the child killed. Solomon uses this difference of preferences to find out who the real mother is and gives her the child.

But what if Anna is the real mother and Bess plays *strategically*, pretending that she too would give up the child? Then Solomon would be in a dilemma. But Solomon was also Machiavellian, since he did not reveal to the two women their actual utilities. They did not know that once they made their choices, Solomon was going to give the child to the woman who offered to give it up. And so the thought of imitating Anna did not occur to Bess.

The Gibbard-Satterthwaite theorem (Gibbard 1973, Satterthwaite 1975, Brams and Fishburn, to appear, Benoit 1999) deals with similar problems. What they have shown is that any social choice function which accepts as input the preference orderings of the agents (voters), and outputs a social preference ordering for the society, will be vulnerable to strategic voting. In other words, there will be situations where some voter i will do better by lying, and giving in as input an ordering which is not her actual ordering, than she would have if she did give her actual ordering as input.

An example from the recent US election shows this. If a voter's actual preferences are Nader, Bush, Gore, Buchanan, in that order, (there were such voters) she would have done better voting for Bush (not her first choice) than for Nader who was her actual first choice. However, using such a strategy required that the voter *knew* how others were voting. In this case the voter who voted for Bush rather than Nader needed to know in advance that almost all other voters were for Bush or Gore and that *they* were not planning to vote strategically. Perhaps strategic voting can be prevented by simply making polls illegal and keeping voters in the dark so as to keep them honest! But this is murky territory and I shall not venture further into it.

Some general considerations: It is time now to consider some general issues which are raised by these examples.

The examples with the carousel and the horses, do show that a social procedure which is viable as an algorithm may not be viable when game theoretic considerations come in. Thus it is in everyone's interest that visibility be maintained at carousels. It is also in the two horsemen's interest that the determination of the slower horse be done speedily. Non-game theoretic algorithms obviously exist for achieving these two goals. But as we saw with the horsemen, not every such algorithm may be viable as a game, for the best thing for a player to do at a game stage, in terms of that player's current utilities, may be something different from what the algorithm calls for.

However, transformations of algorithms can sometimes be found so that the action which the transformed algorithm calls for from some player, turns out to be the same as the action which is in the interests of the player to perform. And occasionally social pressure, or possible legal sanctions, can take up some of the slack. Thus social pressure is needed to keep the passengers from approaching the carousel prematurely. The function of my proposed orange line is merely to increase the social pressure and to make violation of the code obvious. In the case of the two horsemen, however, there is no need of social pressure. What the horsemen should do, namely urge on the horse they are sitting on, and what they want to do, coincide once the horses are switched.

The designer of the social algorithm may be Machiavellian or honest. A Machiavellian designer will encourage players to carry out the desired moves by deceiving them as to the utilities of such moves. The example I gave earlier where the actual payment of $100 is masked by the gambit payment of $10, is of this sort. An honest designer will see to it that the players are fully informed. Such a strategy is more stable, for the algorithm will work even if players find out something which was not expected to be known by them.

Proving programs correct:

Before we consider the problem of proving correctness of games, let us look quickly at an example of the use of Hoare logic for proving a program correct (Kozen and Tiuryn 1990, Cousot 1990, Parikh 2001)

Consider the program α_g for computing the greatest common divisor $gcd(u, v)$ of two positive integers u, v. We want this program to have the property

$$\{u > 0, v > 0\}\ \alpha_g\ \{x = gcd(u, v)\}$$

This property has three components: the *pre-condition* $u > 0, v > 0$, the desired *post-condition* $\{x = gcd(u, v)\}$ and the program α_g itself. We want that if we start with two integers u, v which are both positive, then the program α_g when it ends will have set x to the *gcd* of u, v. One such program α_g, *Euclid's algorithm*, is given by

$$(x := u); (y := v); (\text{while } (x \neq y \text{ do } (\text{if } x < y \text{ then } y := y - x \text{ else } x := x - y))).$$

The program sets x to u and y to v respectively, and then repeatedly subtracts the smaller of x, y from the larger until the two numbers become equal. If u, v are positive integers then this program terminates with x = *gcd*(*u*,*v*), the greatest common divisor of u, v.

The reason is that after the initial (x:=u);(y:=v) clearly $gcd(x, y) = gcd(u, v)$. Now it is easy to see that both the instructions $x := x - y$ and $y := y - x$ leave the *gcd* of x, y unchanged. Thus if B is $gcd(x, y) = gcd(u, v)$, and β is ($if\ x < y\ then\ y := y - x\ else\ x := x - y$), then $\{B\}\beta\{B\}$ holds. Thus if the program α_g terminates, then by Hoare's rule for "*while*," $x \neq y$ will be false, i.e. $x = y$ will hold, and moreover B will hold. Since $gcd(x, x) = x$, we will now have $gcd(u, v) = gcd(x, y) = gcd(x, x) = x$.

Hoare's rules allow us to derive the properties of complex programs from those of simpler ones. Given program β let $\alpha =$"while A do β". A, B are first order formulas. then the Hoare rule says that if we know $\{B\}\beta\{B\}$ i.e. that β preserves the truth of B,[10] then we can conclude $\{B\}\alpha\{B \wedge \neg A\}$. I.e. that α will also preserve B and will moreover falsify A.

In other words, provided that β preserves the truth of B and that B holds when α begins, then B will still hold when α ends and moreover, A will be false. This rule allows us to predict that when the *gcd* program terminates, $gcd(u, v) = gcd(x, y)$ will still hold and $x \neq y$ will be false, i.e. x will equal y.

Hoare Logic was extended by Vaughan Pratt to *Dynamic Logic* which treats programs as modalities. In Pratt's notation, if B is a formula and α is a program, then $[\alpha]B$ is a new formula which says that *after* α is performed, the property B will hold. Hoare's assertion $\{A\}\alpha\{B\}$ can then be rewritten $A \rightarrow [\alpha]B$, if A is true *now*, then *after* α is performed, B will hold.

[10] Indeed it is sufficient if β, applied once, yields B under the precondition that both A and B hold.

However, just as programs can be seen as modalities, so can games. And this insight leads us to our next topic.

Game logic and cake cutting: *Game Logic* is a sublogic of *Dynamic Logic*, i.e. has fewer validities, but enjoys wider semantics. We saw that Dynamic Logic is an extension of Hoare Logic and can be used to prove correctness of programs. Now a program can be thought of as a *one person* game without opponents. Game logic extends this to proper (at first two person) games.

The intuition is to think of a game as a predicate transform. Given a game G and a set of winning positions X, we can define a set Y of starting positions which are favourable to player I, i.e. Y is just the set of positions from which, playing the game G, player I can ensure reaching the set X regardless of player II's actions. Then $Y = G(X)$ as a function of X is monotonic in X. I.e. if $X \subseteq X'$ then $G(X) \subseteq G(X')$. For obviously, if we make it easier for player I to win, we increase the set of positions from which she can start and go on to win. Game logic uses this mathematical fact to good purpose and studies complicated games. We compose games from simpler games, and also compose sets to form more complex sets. Moreover, each game, as we saw, forms a map from sets to sets. Formal details follow. For a full treatment see (Parikh 1983, 1985).

Syntax: Finite number of atomic formulas, P, P_1, Q...etc

Finite number of atomic games: $g_1, ..., g_n$

Formulas: Each P_i is a formula
If A, B are formulas, so are $\neg A$, $A \vee B$
If A is a formula and α is a game, then $(\alpha)A$ is a formula.

Roughly, $(\alpha)A$ means that player I has a winning strategy to play the game α in such a way that A will be true when the game finishes.

Games: Each g_i is a game.
If α, β are games, then so are $\alpha; \beta$, $\alpha \cup \beta$, $< \alpha^* >$, and α^d
If A is a formula, then $< A? >$ is a game.

Here the complex games are seen as composed of simpler ones in an intuitive way. E.g. the game $\alpha; \beta$ is just the composite game which consists of playing first α and then β. $< \alpha^* >$ is α played finitely many times, with player I deciding when to stop. In the composite game $\alpha \cup \beta$, player I decides which of the two games α, β to play. Finally, α^d is just α with the two players interchanged.

Semantics of Game Logic: Given, a set W of states and for each P_i a subset $\pi(P_i)$ of W. For each g_i a set $\rho(g_i) \subseteq W \times \mathcal{P}(W)$. ρ is monotonic so that if $(s, X) \in \rho(g)$ and $X \subseteq Y$ then $(s, Y) \in \rho(g)$. Let $\rho(g)(X) = \{s | (s, X) \in \rho(g)\}$. Then $\rho(g)$ is a function from $\mathcal{P}(W)$ to itself.

Now let

$\pi(A \vee B) = \pi(A) \cup \pi(B)$
$\pi(\neg A) = W - \pi(A)$
$\pi((\alpha)A) = \rho(\alpha)(\pi(A))$

We let $\rho(\alpha;\beta)(X) = (\rho(\alpha)((\rho(\beta)(X))$
$\rho(\alpha \cup \beta)(X) = \rho(\alpha)(X) \cup \rho(\beta)(X)$
$\rho(< \alpha^* >)(X) = \mu Y(X \subseteq Y \wedge \rho(\alpha)(Y) \subseteq Y)$
(where the minimization is relative to set inclusion)
$\rho(< \alpha^d >)(X) = W - \rho(\alpha)(W - X)$
$\rho(< A? >)(X) = \pi(A) \cap X$

A valid formula is one which is true at all points in all models, i.e. a formula A such that $\pi(A)$ is always W.

Complete axiomatization for the valid formulas of the dual free part of Game Logic:

1. All tautologies (or a complete subset of them)

2. $(\alpha;\beta)A \leftrightarrow (\alpha)(\beta)A$

3. $(\alpha \vee \beta)A \leftrightarrow (\alpha)A \vee (\beta)A$

4. $(< \alpha * > A) \leftrightarrow (A \vee (\alpha)(< \alpha * > A))$

5. $(< A? >)B \leftrightarrow A \wedge B$

Rules:

1. Modus ponens: from $A, A \rightarrow B$ derive B.

2. Monotonicity: from $A \rightarrow B$ derive $(\alpha)A \rightarrow (\alpha)B$

3. Bar induction: from $(\alpha)A \rightarrow A$ derive $(< \alpha^* >)A \rightarrow A$

This set of axioms and rules is complete. A sound and possibly complete axiom for the dual operator is

$(\alpha^d)A \leftrightarrow \neg(\alpha)(\neg A)$

Cake cutting: In cake cutting the number of players goes up to n. The cake cutting algorithm due to Banach and Knaster goes as follows. If there are n players, then the first player cuts out a slice which she claims is her fair share. Then the other players examine it in turn. Anyone who thinks it is too big may reduce it and put something back in the main cake. After each one has examined the slice, the last person to have reduced it, takes it. If no one reduced it, then the person who cut the slice takes it. At this stage we have $n - 1$ players left and the procedure is repeated. The following questions arise:

What does it mean to say that the cake cutting algorithm is fair?

How do we prove that it is fair?

The answer is that each player has a winning strategy to ensure $\mu_i(P_i) \geq 1/n$, where P_i is the piece received by player i, and μ_i is player i's personal measure by which she evaluates the value of a piece. It is assumed that the value of the whole cake is 1. We skip the details of the proof of correctness, which will be found in (Parikh 1983, 1985). The proof makes use of stability properties analogous to that we saw in the *while* rule of Hoare.

To give just one example of the techniques used in the proof, let r be the move which reduces the slice, i.e. puts something back from the slice to the main part of the cake. This is an action which occurs repeatedly in the algorithm. And let $F(m, k)$ be the proposition that the main part m of the cake is large enough for k people. Then $F(m, k)$ is invariant under the action r. If $F(m, k)$ is true before r, then it is still true after. Another property which holds is that if $F(m, k + 1)$ is true then it *can* be true that after the action c, of cutting a slice from the cake, the remaining main part is big enough for k people *and* the slice itself is big enough for one. Note the contrast, that the action r preserves $F(m, k)$ *regardless* of how r is performed whereas the action c, of cutting the cake, *may* yield the desired outcome but need not.

Future Work: Our basic structure, as in the original work in Game logic will remain that of a state space W where to each state is assigned a unique player whose turn it is to act. Moreover at each state $s \in W$, the acting player has a choice of (deterministic) actions.

We will also assume utility functions for the agents assigned to the various states. And our assumption will be that at each state, the agent chooses an action which will maximize the utility of the resulting state. However, we will not make *any* assumption of mutual knowledge of rationality. Nor will we assume as part of our model that the agents have any theory of the other agents or any knowledge of their rationality.

This procedure will have two advantages. One is that it simplifies our work and makes it more general since it will also apply to games where the agents do not know each other's utilities. The other is that it will be more realistic since we know in fact that real agents reason about other agents only in a partial and haphazard way. Building in 'mutual knowledge of rationality' assumptions into our model would exlude consideration of such situations and would moreover involve us in curently active controversies which are not central to this paper.

There are various ways a game designer can achieve the requisite conditions on utilities. One extreme is to observe every action performed by every agent and impose penalties on performance of undesired actions. Thus for example we could prevent speeding by having every car be accompanied by a police car. This is unrealistic, but is not excluded by purely theoretical considerations. Something like this does indeed happen in examination halls where non-cheating is enforced through diligent supervision.

At the other extreme, we may impose certain utilities only at boundary points and rely on the agents to calculate the utilities of intermediate points both for themselves and for the other game players. For example, in chess, the {win, lose, draw} trichotomy is imposed only at the end of the game and the notions of 'good position' or 'bad move' are derived notions. Locally, each player tries to make his own position 'better' in terms of whatever criterion is being used. Globally, each player is trying to win. If the players were logically omniscient, then of course there would be a tight connection between the possibility of winning (or at least not losing) and the local notion of goodness. However, if the players' logical powers are limited, the connection is a rough one, not easy for a mathematical theory to flesh out.

With cake cutting, there is no social goal as such. Each player has his own goal and the structure of the game ensures that each player can achieve his goal regardless of the actions of the other players. With other games the situation may be more complex. For example we may impose fines for going through a red light. These fines then create a slight predisposition to obey the lights. Once there is a slight bias in favour of obeying the traffic signal, the risk of an accident (because of the behaviour of other motorists) becomes a strong force which reinforces the bias introduced by fines.

Development of a formal model which takes these factors into account will be carried out in future work. The reader may feel that she has seen a great many loosely connected, but markedly different themes. I plead guilty to this charge. The richness of social software is so great that giving a survey of the range of possibilities is a necessary first step.

Pointers to other work: The two books by Brams and others are very much worth reading as they are both instructive and entertaining. Prashant Parikh (2000) gives an application of game theory to show how ambiguities in normal discourse are resolved through Nash equilibria. Elections are an extremely important and well studied case of social software. Paradoxically, the Arrow theorems seem to indicate that elections are a software for which a specification does not exist. Reasonable conditions we could impose, on what a social choice function should look like, turn out to be incompatible. However, there are occasions when we do know what we want an election to achieve.

One goal that has been considered is that whenever there is a Condorcet winner, one who would beat any other candidate, then the Condorcet winner should be the choice of the election procedure. In fact even this is not quite clear. Suppose we have two candidates A and B, where A commands 51% of the vote but is strongly disliked by the other 49% and B is rather liked by 90%. Then A might well beat B in a head to head contest, but B should be the preferable candidate. The survey by Brams and Fishburn, and the book by Dummett are good introductions to the theory of elections. Pauly contructs a logic for the study of coalitions and proves completeness. Our own paper, (Parikh 2001) gives a direction for understanding language which replaces the normal role played by truth by a social software paradigm which makes many puzzles, e.g. Quine's indeterminacy of translation, much clearer. Finally, an immense debt is owed to Wittgenstein, whose language games have been a large part of the inspiration for this paper.

Other work which is related to some of the issues includes certainly the path breaking paper by Aumann (1976) with sequels by Geanakoplos and Polemarchakis, and others, as well as one by Krasucki and myself (Parikh and Krasucki 1990) in which the issue of revision of beliefs under communication and the possibility of common knowledge are studied. The book by Fagin et al (1995) is an important resource for knowledge issues. The paper by Brandenburger and Keisler (2000) presents an interesting paradox which arises when we consider two agents and what mutual beliefs they might have about each other. Finally, we should acknowledge the debt to the classic books by Hintikka (1962) and Lewis (1969) which investigate knowledge and co-ordination respectively. I apologize to all the other very important sources which I have not listed – listing them all would be a paper by itself.

Acknowledgements We thank Johan van Benthem, Arthur Collins, Horacio Arlo Costa, Walter Dean, Amy Greenwald, Bud Mishra, Larry Moss, Prashant Parikh and Marc Pauly for comments. The pointer to the Moore survey (1992) was given to us by Roberto Serrano. Previous versions of this paper were delivered in Hyderabad, India, in Prague, at NYU, at Brown University, at the University of Maryland, College Park, at the Albert Meyer Symposium at Boston University, and at the University of Helsinki. We thank the audiences for the comments.

References:

Arthur, W.B.,"Inductive reasoning and bounded rationality", *Complexity in Economic Theory*, **84** 1994, pp. 406-411.

Aumann, R., "Agreeing to disagree", *Annals of Statistics* **4** 1976, 1236-1239.

Benoit, J.P., "Strategy proofness and when ties are permitted", research report, Economics Department, NYU (1999).

Benthem, Johan van, "Logic and procedure in legal reasoning", to appear.

Benthem, Johan van, "When are two games the same?", to appear. Presented at *LOFT-3*, Torino 1998.

Bicchieri, Cristina, *Rationality and Coordination*, Cambridge U. Press, 1984.

Brams S., *Theory of Moves*, Cambridge U. Press, 1994.

Brams, S., and A. Taylor, *Fair division*, Cambridge U. Press, 1996.

Brams, S., and P. Fishburn, "Voting procedures", in *Handbook of Social Choice and Welfare*, edited by Arrow et al, Elsevier, to appear.

Brandenburger, A., and J. Keisler, "An impossibiity theorem on beliefs in games", 1999.

Chopra, C., and R. Parikh, An Inconsistency Tolerant Model for Belief Representation and Belief Revision appeared in *Proc. IJCAI 99.* Full version to appear in *Annals of*

Math and AI.

Chwe, M., *Rational Ritual: Culture, Coordination and Common Knowledge*, Princeton U. Press, 2001.

Cousot, P., "Methods and logics for proving programs", in *Handbook of Theoretical Computer Science*, vol B, edited by J.V. Leeuwen, North Holland 1990, pp. 841-994.

Dummett, M.A.E., *Principles of Electoral Reform* , Oxford University Press, 1997.

Fagin, R., J. Halpern, Y. Moses and M. Vardi, *Reasoning about Knowledge*, MIT press, 1995.

Geanakoplos, J., and Polemarchakis, H., "We can't disagree forever", *J. Economic Theory* **28** (1982) 192-200.

Gibbard, A., "Manipulation of voting schemes: a general result", *Econometrica* **41** (1973) 587-601.

Greenwald, A., B. Mishra and R. Parikh, "The Santa Fe bar problem revisited" presented at the Stony Brook workshop on Game theory, summer 1998.

Hintikka, J., *Knowledge and Belief*, Cornell U. press 1962.

Kozen, D., and R. Parikh, "An Elementary Completeness Proof for PDL", *Theor. Comp. Sci* 14 (1981) 113-118.

Kozen, D., and J. Tiuryn, "Logics of Programs", in *Handbook of Theoretical Computer Science*, vol B, edited by J.V. Leeuwen, North Holland 1990, pp. 789-840.

Lewis, D., *Convention: A Philosophical Study*, Harvard U. press 1969.

Moore, J., "Implementation, contracts, and renegotitation in environments with complete information", in *Advances in Economic Theory, Sixth World Congress*, Cambridge U. Press 1992.

Parikh, P., "Communication, meaning, and interpretation", *Linguistics and Philosophy*, volume 23, number 2, April 2000 pp. 185-212

Parikh, R., "Propositional logics of programs: new directions", in *Foundations of computation theory*, edited by Marek Karpinski, Springer *Lecture Notes in Computer Science* #158 (1983) pp. 347-359.

Parikh, R., "The Logic of games and its applications", *Annals of Discrete Math.*, 24 (1985) 111-140.

Parikh, R., "Vagueness and Utility: the Semantics of Common Nouns", in *Linguistics and Philosophy* **17** 1994, 521-35.

Parikh, R., "Language as social software" (abstract), International Congress on Logic, Methodology and Philosophy of Science (1995), page 417. Full paper in *Future Pasts: the Analytic Tradition in Twentieth Century Philosophy*, Ed. J. Floyd and S. Shieh, Oxford U. Press, 2001, 339-350.

Parikh, R., and P. Krasucki, "Communication, Consensus and Knowledge", *J. Economic Theory* **52** (1990) pp. 178-189.

Parikh, R., L. Parida and V. Pratt, "Sock sorting: an example of a vague algorithm", to appear in *Logic J. of IGPL*, (2001)

Parikh, R., and R. Ramanujam, "Distributed processing and the logic of knowledge", in *Logics of Programs*, Proceedings of a Conference at Brooklyn College, June 1985, Springer Lecture Notes in Computer Science #193.

Pauly, M., "An Introduction to game logic", in *Fromalizing the Dynamics of Information Flow*.

Pauly, M., "A Modal logic for coalitional power in games", to appear.

Pratt, V., "Semantical considerations in Floyd-Hoare Logic", in *Proc. 17th Annual IEEE Symposium on Foundations of Computer Science*, (1976) 109-121.

Satterthwaite, M., "Strategy-proofness and Arrow's conditions", *J. Economic Theory* **10** (1975) 187-217.

Wittgenstein, L., *Philosophical Investigations*, Tranlsated by G.E.M. Anscombe, Basil Blackwell 1953.

Wittgenstein, L., *Remarks on the Foundations of Mathematics*, edited by G.H. von Wright, R. Rhees, G.E.M. Anscombe, MIT press 1978.