2002

TR-2002005: Towards a Theory of Social Software

Rohit Parikh

Towards a Theory of Social Software

Preliminary version - not for circulation
To be presented at *Deon'02*

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]

Is it possible to create a theory of how social procedures work with a view to understanding them and designing better ones? I want to start with an illustrative example of a common difficulty.

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.

[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.

	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 fooling 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.

The Recommendation Letter: 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.

Admission Decisions: A similar difficulty arises with the selection of students for admission to college. Suppose that prestigious university PU wants to choose between two students A, B who have gone to different schools. Not wanting to rely entirely or primarily on scores obtained on some multiple choice test, they put a heavier emphasis on grades. But now consider the teachers T_a, T_b of the students A, B. T_a naturally wants *her* student A to get in and she can help by giving him a higher score than he deserves. Whatever pangs of conscience she might feel at this will be assuaged by the thought that probably teacher T_b is already giving a higher score to B than he deserves, and it would not be fair to A to give him a more realistic score, it would only penalize him. Teacher T_b will have similar thoughts and thus both students will get higher scores than they deserve.

Is there a solution? Surely university PU could create a database of scores obtained by students at various schools and the eventual performance of these

students. Schools which habitually give out high score could be downrated, thereby restoring some measure of objectivity.

The Election: For yet another situation which is related consider the case of an election with three candidates A, B, C and three voters a, b, c. The election will consist of two phases. In the first phase, A will be matched against B. After that, the winner of the first election will run against C.

The three voters a, b, c have preferences, respectively: $A > B > C$ for a, $B > C > A$ for b and $C > A > B$ for c. Since two out of three voters prefer A to B, A will win over B. After that, A will run against C and since both b and c prefer C to A, C will be elected. This is not an outcome that a is likely to be happy with. However, suppose that a pretends that his preferences are actually $B > A > C$ and votes for B against A in the first round. In that case B will win the first election and go on to win against C as well.

Thus by pretending that his preferences are different from what they are, voter a ensures a better outcome for himself.

The Gibbard Satterthwaite theorem says that such situations, where voters benefit by voting differently from their actual preferences, are endemic to all electoral systems. Indeed, in the last presidential election in the US, people whose first preference was Nader would have done better to vote for Gore instead. As it is, by voting for Nader, they ensured the election of Bush who was (for most Nader voters) a lower preference than Gore.

As we saw, 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.

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 pro-

grams 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 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. We start with a discussion of *one way functions* which are common in Computer Science but are less used in real life.

One way functions: 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 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. A commonly used one way function in cryptography multiplies two large primes p, q into a composite number $n = p \times q$. Give p, q it is easy to compute n. However, even though p, q are uniquely defined by n, they are, as far as we know, extremely difficult to compute.

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 first

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.[2]

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.

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 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. One device suggested by Greenwald *et al* is a system of taxes imposed on the bar-goers,

[2]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.

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.

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

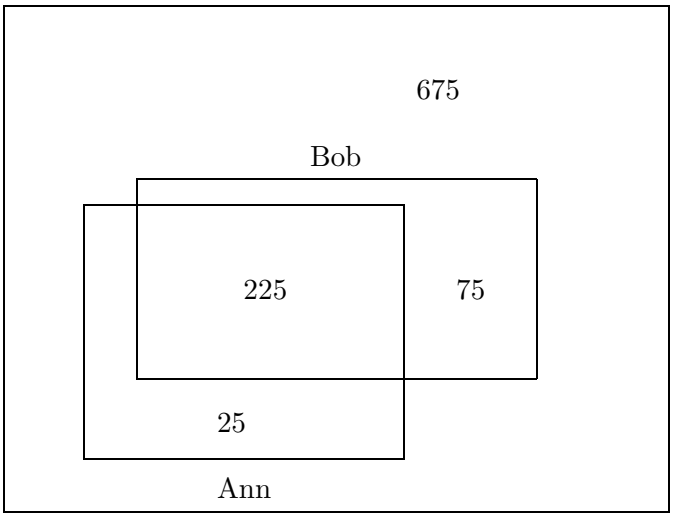
> If I tell someone "Stand roughly there" – may not this explanation work perfectly? And cannot every other one fail too?
>
> *But isn't it an inexact explanation?* – Yes; why shouldn't we call it "inexact"? Only let us understand what "inexact" means. For it does not mean "unusable".

I would like to suggest that a degree of exactness is sufficient if it is adequate for the particular social algorithm.

Vague predicates 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 shade gradually into cases where it clearly does not. Social algorithms have to provide for these difficulties too. As has been remarked by Berlin and Kay, there is substantial difference between the way in which two different individuals assign color terms to objects. A patch may be described as purple by one viewer and as blue by another. This *could* lead to difficulties.

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).

(Parikh 94) also gives an example of a situation where two people assign different extensions to the word 'blue' and yet when one of them describes a book as blue to the other one, this actually helps in that the searching time is reduced. In this example, Ann is helping Bob to find a particular book, say a topology book, from among a thousand. Ann describes the topology book as *blue*, but in fact Ann and Bob assign somewhat different extensions to the word 'blue'. It still turns out that Bob's searching time is considerably reduced by Ann's information.



In the picture above, of the 1,000 books, there are 250 which Ann would classify as blue and 300 which Bob would classify as blue, with an overlap of 225 books which both would so classify, and 675 which neither would so classify. Without Ann's description, Bob must look through all books. *With* her description "It's blue", Bob would look first through the books that he thinks are blue and there is a 90% chance that Ann's book is among them. Thus his searching time is much reduced.

We now proceed to describe a particular formal theory for proving the correctness of (some) social algorithms.

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.

The results on cake cutting and multi modal logic which we have referred to above have recently been extended by Marc Pauly who has formalized a logic for coalitions.

References:

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

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.

Berlin, Brent, and Paul Kay, *Basic color terms; their universality and evolution*, University of California Press (1969)

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.

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.

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

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.

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

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

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., "Social software" to appear in *Synthese*, (2002)

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

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*, Translated by G.E.M. Anscombe, Basil Blackwell 1953.