

Knowledge and Structure in Social Algorithms

Rohit Parikh

Brooklyn College and CUNY Graduate Center
City University of New York
365 Fifth Avenue
New York, NY 10016
<http://sci.brooklyn.cuny.edu/cis/parikh/>

Abstract. While contemporary Game Theory has concentrated much on strategy, there is somewhat less attention paid to the role of knowledge and information transfer. There are exceptions to this rule of course, especially starting with the work of Aumann [2], and with contributions made by ourselves with coauthors Cogan, Krasucki and Pacuit [16, 12]. But we have still only scratched the surface and there is still a lot more that can be done. In this paper we point to the important role which knowledge plays in social procedures (colorfully called *Social Software* [14])

Key words: Knowledge, Society, Algorithms

The peculiar character of the problem of a rational economic order is determined precisely by the fact that the knowledge of the circumstances of which we must make use never exists in concentrated or integrated form, but solely as the dispersed bits of incomplete and frequently contradictory knowledge which all the separate individuals possess. The economic problem of society is thus not merely a problem of how to allocate "given" resources – if "given" is taken to mean given to a single mind which deliberately solves the problem set by these "data." It is rather a problem of how to secure the best use of resources known to any of the members of society, for ends whose relative importance only these individuals know.

F. Hayek

Individualism and Economic Order

1 Introduction

The first third of the XXth century saw two important developments. One of these was Ramsey's tracing of personal probabilities to an agent's choices [21]. This was a precursor to the work of de Finetti, von Neumann and Morgenstern, and Savage [5, 9, 22]. The other one was Turing's invention of the Turing machine

[24] and the formulation of the Church-Turing thesis according to which all computable functions on natural numbers were recursive or Turing computable.¹

Game theory has depended heavily on the first of these developments, since of course von Neumann and Morgenstern can be regarded as the fathers of Game theory. But the other development has received less attention. That development led to the development and design of computers and also to fields like Logic of Programs, Complexity Theory and Analysis of Algorithms. It also resulted in much deeper understanding of algorithms, but only of computer algorithms. Social algorithms have remained largely unanalyzed mathematically except in special subfields like Social Choice Theory [1] or Fair Division [3]. These fields, however, do not tend to analyze *complex* social algorithms (even algorithms of modest complexity like the two thousand year old Euclid's algorithm) as is done in computer science.² The typical game theoretic example tends to be either a one shot game, or else such a game repeated.

A later development, going back to the work of Hintikka, Lewis and a little later Aumann [7, 8, 2], brought in the issue of *knowledge*. The notion of common knowledge is of course very important for Aumann as *common knowledge of rationality* can be seen as a justification for backward induction arguments.

But knowledge too has received less attention than it might. We all know that the Valerie Plame affair [20] had something to do with someone knowing something which they should not have, and someone revealing something which they should not have. But why should they not? Clearly because of certain possible consequences. Knowledge and knowledge transfer are ubiquitous in how social algorithms work. Note that the fact that the FBI bugged Burriss's phone conversations with Blagojevich's brother played an important role, and the fact that we do not want the FBI to have unlimited right to listen in on conversations are extremely important knowledge considerations.

We will try in this paper to bring attention to the importance of the two issues of knowledge and logical structure of algorithms, and show the way to a broader arena in which game theorists might want to play. Hopefully, in fact almost certainly, there is a rich general theory to be developed.

¹ The research reported here was supported in part by a research grant from the PSC-CUNY program. Previous versions of this paper were presented at a workshop on knowledge at the University of Quebec in Montreal (2007), and at the World Game Theory meeting at Northwestern University (2008)

² But society itself is replete with extremely complex algorithms. Just consider the complexity involved in Obama's election to the presidency, the consequent vacating of his senate seat, Blagojevich's acquiring the right to name Obama's successor, Blagojevich naming Burriss to Obama's vacant seat, Blagojevich's impeachment and removal from office, demands, so far unsuccessful, for Burriss to step down, and, no doubt, quiet satisfaction on the part of the Republicans. And even Obama's election to the presidency is hardly a simple event since it involved factors like Hillary's association with her husband, a former president, an initial feeling on the part of African-Americans that Obama, having no ancestry in the institution of slavery was not "one of us," etc. etc.

The notion of algorithm is implicit in so many things which happen in everyday life. We humans are tool-making creatures (as are chimps to a somewhat smaller extent) and both individual and social life is over-run with routines, from cooking recipes (Savage’s celebrated eggs to omelette example [22] for instance) to elections – a subject of much discussion going back to Condorcet.

Over the last ten years or so, a field called *Social Software* [14] has come into existence which carries out a systematic study of such questions, and the purpose of this paper is to give an introduction to the knowledge-theoretic issues. We will proceed by means of examples.

2 Structure

Normally, a piece of social software or social algorithm has a logical structure. As was argued in [14], this structure must address three important aspects, namely incentives, knowledge, and logical structure. For normally, an algorithm has logical structure, “ A happens before B , which is succeeded by action C if condition X holds and by action D if X does not hold.”

But quite often, the logical structure of the algorithm is parasitic on logical (or algorithmic) properties of existing physical or social structures. Clearly a prison needs a certain physical structure in order to be *able* to confine people, and a classroom needs a blackboard or a lectern in order for it to be usable as the venue of a lecture. Thus the teacher can now perform actions like write “No class tomorrow” on the blackboard and the students can read what she wrote or copy it in their notebooks. The physical properties of the blackboard enable certain actions with their own algorithmic properties. The fact that there is no class the next day now becomes common knowledge and the students can make plans to use the time that has been freed up.

2.1 Queues

A *social structure* with certain logical properties is a queue.

The queue is a very popular institution which occurs both in daily life and in computer programs. In a computer program, a queue is a FIFO structure, where FIFO means, “First in, first out.” There are two operations, one by which an element is deleted from the front of the queue, and a second one where an element is added to the back of the queue. In real life, the queue could consist of people waiting at a bank to be served. The person *deleted* is the one who was at the front of the queue but is no longer in the queue, and who receives service from a teller. An element which is *added* is a new customer who has just arrived and who goes to the back of the queue.

Clearly the queue implements our notions of fairness, (which can be proved rigorously as a theorem) that someone who came earlier gets service earlier, and in a bank this typically does happen. If someone in a bank tries to rush to the head of the line, people will stop him. Thus ‘*violations easily detectable*’ is a crucial knowledge property.

We also have queues at bus stops and quite often the queue breaks down; there is a rush for seats at the last moment. Presumably the difference arises because things happen much faster in a bus queue than they do in a bank. At a bus stop, when the bus arrives, everything happens very fast and people are more interested in getting *on* the bus than in enforcing the rules.

Consider now, by comparison, the problem of *parking*, which is a similar problem. A scarce resource needs to be allocated on the basis of some sort of priority, which, now, is more difficult to determine. When people are looking for parking in a busy area, they tend to cruise around until they find a space. There is no queue as such, but in general we still want that someone who arrives first should find a parking space and someone who arrives later may not. This is much more likely in a university or company parking lot, which is compact, and may even have a guard, rather than on the street, where parking is distributed, and priority does play *some* role but it is only probabilistic. Clearly the lack of information about where the parking space is, and who came first, plays an important role.

This fact has unfortunate consequences as Shoup [23] points out.

When my students and I studied cruising for parking in a 15-block business district in Los Angeles, we found the average cruising time was 3.3 minutes, and the average cruising distance half a mile (about 2.5 times around the block). This may not sound like much, but with 470 parking meters in the district, and a turnover rate for curb parking of 17 cars per space per day, 8,000 cars park at the curb each weekday. Even a small amount of cruising time for each car adds up to a lot of traffic.

Over the course of a year, the search for curb parking in this 15-block district created about 950,000 excess vehicle miles of travel – equivalent to 38 trips around the earth, or four trips to the moon. And here’s another inconvenient truth about underpriced curb parking: cruising those 950,000 miles wastes 47,000 gallons of gas and produces 730 tons of the greenhouse gas carbon dioxide. If all this happens in one small business district, imagine the cumulative effect of all cruising in the United States.

Shoup regards this problem as one of incentive and suggests that parking fees be raised so that occupancy of street parking spaces is only 85%. But clearly this will penalize the less affluent drivers. The new fees will likely be still less than the cost of garage parking, affluent drivers will abandon garage parking for street parking, and the less affluent drivers will be priced out. Note by contrast that we do not usually charge people for standing in a queue. We could, and surely queues would also be shorter if people had to pay to stand in them. But this has not occurred to anyone as a solution to the ‘standing in line problem.’

An algorithmic solution to the problem of parking might well be possible using something like a GPS system. If information about empty parking spaces was available to a central computer which could also accept requests from cars for parking spaces, and allocate spaces to arriving cars, then a solution could in fact be implemented. The information transfer and the allocation system would

in effect convert the physically distributed parking spaces into the algorithmic equivalent of a queue. There would be little wasteful consumption of gasoline, and the drivers would save a great deal of time and frustration.

And here indeed is an implementation of the alternate solution

Find a Place to Park on Your GPS – Spark Parking Makes it Possible

Navigation Developers Can Access Spark Parking Points of Interest Through New Tele Atlas ContentLink Program

San Francisco, CA, March 21, 2007

Running late for a meeting and worried about finding a place to park? Unhappy about paying outrageous valet parking fees at your favorite restaurant? These headaches will soon be a thing of the past. Spark Parking's detailed parking location information data is now available through the newly released Tele Atlas ContentLinkSM portal for application developers to incorporate into a range of GPS devices and location-based services and applications.

Spark Parking's detailed parking information provides the locations of every paid parking facility in each covered city – from the enormous multi-level garages to the tiny surface lots hidden in alleys. In addition, Spark Parking includes facility size, operating hours, parking rates, available validations, and many more details not previously available from any source. As a result, drivers will easily be able to find parking that meets their needs and budgets.

<http://www.pr.com/press-release/33381>

SAN FRANCISCO

Where's the bus? NextMuni can tell you.

System uses GPS to let riders know when streetcar will arrive

Rachel Gordon, Chronicle Staff Writer

Thursday, March 29, 2007

San Francisco's Municipal Railway may have a hard time running on time, but at least the transit agency is doing more to let riders know when their next bus or streetcar is due to arrive.

The "NextMuni" system, which tracks the location of vehicles via satellite, is now up and running on all the city's electrified trolley bus lines. It had been available only on the Metro streetcar lines and the 22-Fillmore, a trolley bus line that served as an early test.

The whereabouts of the Global Positioning System-equipped vehicles are fed into a centralized computer system that translates the data into user-friendly updates available on the Internet and on cell phones and personal digital assistants.

<http://www.sfgate.com/>

Ultimately, the difference between queues and searching for parking is structural. In one case there is an easy algorithmic solution which respects priority (more or less) and in the other case such solutions are harder to find – except when we are dealing with parking lots or use sophisticated new technology.

2.2 Keys

Here is another example. When you rent an apartment, you receive a key from the landlord. The key serves two purposes. Its possession is *proof of a right*, the right to enter the apartment. But its possession is also a *physical enabler*. The two are not the same of course, since if you lose your key, you still have the *right*, for it is still your apartment. But you are not *enabled*, as you cannot get in. If some stranger finds the key, then he is enabled, but does not have the right. Thus the two properties of a key do not coincide perfectly. But normally the two do coincide.

There are other analogs of a key which perform similar functions to a key. A password to a computer account is like a key, but does not need to be carried in your pocket. An ID card establishes your right to enter, but you typically need a guard to be present, to see your card and to let you into the building. If the building is locked and the guard is not present, you are out of luck.

In any case, these various generalized keys differ algorithmically in some crucial ways. Stealing someone’s identity was at one time very difficult. You had to look like that person, know some personal facts, and you had to stay away from that person’s dog who knew perfectly well that you had the wrong smell. You needed a different ‘ID’ for the dog than you needed for people.

But nowadays identity theft is extremely easy. Lots of Social Security numbers, and mothers’ maiden names are out there for the taking, and people who do not look like you at all can make use of them. Personal appearance or brass keys which originally provided proof of “right to entry,” have been replaced by electronic items which are very easy to steal.

Let x be an individual, and let $R(x)$ mean that x has the *right* to use the resources controlled by the key, and $E(x)$ mean that x is *enabled* by the key. Then we have two important conditions.

- **Safety:** $E(x) \rightarrow R(x)$. Whoever is enabled has the right
- **Liveness:** $R(x) \rightarrow E(x)$. Whoever has the right is enabled.

Of course safety could be thought of in terms of the contrapositive,

$$\sim R(x) \rightarrow \sim E(x)$$

namely, whoever does not have the right is not enabled. Usually, safety is more important than liveness. If you lose your key and someone finds it, you are in trouble. But liveness also matters. A good notion of *key* must provide for both properties.

At one time, university libraries tended to be open. People not connected to the university, even if they did not have the right, were still *able* to enter the library. There was open access corresponding to the fact that liveness was

thought of as more important than safety. But the trend in the last few decades has been in the opposite direction and entry to libraries is strictly controlled, at least in the US.

In any case the structural problem (of safety) can be addressed at the incentive level, for instance by instituting heavy penalties for stealing identities. But we could also look for a structural solution without seeking to penalize anyone.

Toddlers are apt to run away and get into trouble, but we do not solve the problem by punishing them – we solve it by creating barriers to such escape, e.g., safety gates. A magnetic card which you can swipe also serves as a purely structural solution to the safety problem.

Another interesting example is a fence. A farmer may have a fence to keep his sheep in, and the fence prevents certain kinds of movement – namely sheep running away. Here the fence is a physical barrier and implements the safety condition in a purely physical way. But sometimes, on a university campus, we will see a very low fence around a grassy area. Almost anyone can walk over the fence, so the fence is no longer a physical obstacle. Rather the value of the fence is now informational, it says, *Thou shalt not cross!* With the yellow tape which the police sometimes put up, perhaps around a crime scene, or perhaps simply to block off some intersection, the *Thou shalt not cross* acquires quite a bit of punch.

3 Crime and Punishment

We offer a simple model to explain certain common situations where knowledge plays a role and can be used for reward or punishment.

3.1 Prisoner's dilemma

In this game, two men are arrested and invited to testify against each other. If neither testifies, then there is a small penalty since there is no real evidence. But if one *defects* (testifies) and the other does not, then the defector goes free and the other gets a large sentence. If both defect they both get medium sentences. Jointly they are better off (The payoffs are 3 each) if neither defects, but for both of them, defecting is the dominant strategy. It yields better payoffs regardless of how the other acts. But if they both defect, then they end up with (1,1) which is worse. If one defects and the other remains honest then the honest one suffers for his honesty.

	Coop	Def
Coop	3, 3	0, 4
Def	4, 0	1, 1

There is a unique, rather bad Nash equilibrium at SE with (1,1), while the (3,3) solution on NW, though better for *both*, is not a Nash equilibrium.

Let us change this now into a three person game, where the third agent S (Society) has a payoff equal to the sum of the payoffs of the two original agents.

Consider now the expanded game G. In G, after the first two players make their moves, the third player moves and can choose among p_r (punish Row), p_c (punish Column), p_b (punish both), and n (no action). p_r , as we might expect, results in a negative payoff for Row of say 5. If Row has defected, S can play p_r which results in a negative payoff of 5 for the Row player. Similarly for Column and p_c . G is a full information game in that after Row and Column have made their moves, S knows what moves they made. Since S also suffers when Row or Column betrays his partner, S has an incentive to punish the erring player and the threat of S's punishment will keep the two players honest. We now get (from the point of view of Row and Column)

	Coop	Def
Coop	3, 3	0, -1
Def	-1, 0	-4, -4

and the NE solution with payoffs of (3,3) becomes the unique Nash equilibrium.

The game G' is just like the game G, except that S *lacks information* as to who made which move. If the societal payoff is only 4 or less, S knows that one of Row and Column cheated but it does not know which one. Thus it has no way to punish, and cheating can take place with impunity.

Clearly socially responsible behavior is more likely in G than in G' and the difference arises from the fact that in G, S has some information which it does not have in G'.

This, of course is why the FBI taps the phones of suspected criminals. A social agency has the incentive to punish anti-social behavior, and in order to do this, it needs to get information and change a G'-like situation into a G-like situation.³

Naturally, the agency S might not be benign. S could easily be a Mafia boss who needs to know when some member of the mob "sings", i.e., betrays the oath of silence. The singer could then be punished if and when he comes out of prison.

The FBI could itself have non-benign reasons for tapping phones. For instance we know that Martin Luther King's phone was tapped in order for the FBI to have power over him. This situation can be represented game theoretically, by turning G into a *four payer game* where the FBI like agent (call is S_1) which has the power to punish is not society at large but an *agent* of society, and society, while wishing to control anti-social behavior on the part of Row and Column, also needs to control its own agent whose job it is to keep Row and Column in check but who may have its own payoff function distinct from social welfare.

We shall not go more into this in this paper.

³ Of course all this is rather obvious, but it is important to point to the game theoretic reason not only behind punishment, but behind the acquisition of information relevant to it.

4 Cooperative Knowledge

Distributed Algorithms are much studied by computer scientists. A lot of commercial activity which goes on on the web has the property of being a distributed algorithm with many players. And of course the market is itself a very old distributed algorithm.

In such algorithms, it is crucial to make sure that when agents have to act, they have the requisite knowledge. And models for calculating such knowledge have existed for some time; we ourselves have participated in constructing such models [19, 18]. See also [4].

The notion of *common knowledge* as the route to consensus was introduced by Aumann in [2]. There is subsequent work by various people, including Geanakoplos and Polemarchakis [6] and ourselves [16]. Aumann simply assumed common knowledge, and showed that two agents would agree on the value of a random variable if they had common knowledge of their beliefs about it. [6] showed that even if the agents did not have common knowledge to start with, if they exchanged values, they would arrive at consensus, and common knowledge of that fact. Parikh and Krasucki [16] carried this one step further and considered many agents exchanging values in *pairwise interactions*. No common knowledge could now arise, as most agents would remain unaware of individual transactions they were not a party to. Nonetheless there would be consensus. Thus this exchange of values could be seen as a distributed algorithm which achieved a result.

Issues about how knowledge enters into social algorithms are discussed in [10, 12, 19].

[19] actually discusses how a framework for defining knowledge can be developed. A finite number of agents have some private information to start with, and they exchange messages. Each exchange of messages reveals something about the situation, or, in technical terms, it reduces the size of the relevant Kripke structure or Aumann structure. An agent who has seen some events but not others can make guesses as to what other events *could* have taken place and it knows some fact ϕ iff ϕ would be true *regardless* of how the unseen events went. This framework is used in both [12, 10].

[10] discusses agents who are connected along some graph, and knowledge can move only along the edges of a graph. Thus if agent i is not connected to agent j , then i cannot directly obtain information from j , but might get such information via a third agent k , as in fact Novak got some information from Judith Miller. Such edges may be approved or disapproved, and if information transfer took place along a disapproved edge, then that could be cause for legal sanctions, not because harm had occurred, but because harm *could* occur and the algorithm was no longer secure.

It is shown in [10] that the graph completely determines the logical properties of possible states of knowledge, and vice versa. Indeed, an early version of that paper already discussed the Plame case before it hit the headlines.

In [12] we consider how obligations arise from knowledge. We consider the following examples:

Example 1: Uma is a physician whose neighbour is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbour.

Example 2: Uma is a physician whose neighbour Sam is ill. The neighbour's daughter Ann comes to Uma's house and tells her. Now Uma does have an obligation to treat Sam, or perhaps call in an ambulance or a specialist.

Example 3: Mary is a patient in St. Gibson's hospital. Mary is having a heart attack. The caveat which applied in case a) does not apply here. The hospital has an obligation to *be aware* of Mary's condition at all times and to provide emergency treatment as appropriate.

In such cases, when an agent cannot herself take a requisite action, it is incumbent upon her to provide such information to the agent who *can* take such action. Or, as in the case of the hospital, the agent has an obligation not only to act, but also to gather knowledge so as to be *able to act* when the occasion arises. A milder example of such situations consists of requiring homeowners to install fire alarms. Homeowners are not only required (advised) to take action when there is a fire, they are also required to set up a system such that *if there is a fire, they will know about it*.

Again the semantics from [19] is used. Various possible sequences of events are possible, depending on the actions taken by the agents. Some of these sequences are better than others, and some, indeed, are disastrous, as when a patient is not treated for lack of information. It is shown how *having information* creates obligations on the agents, and also how the need to *convey information* arises, when one knows that an agent who could carry out some required action lacks the requisite information.

5 Summary

We have given examples of situations where knowledge transfer and algorithmic structure can affect or even determine the sorts of social algorithms which are possible. As we have said earlier, understanding the role of knowledge in the working of society is a big project. The importance of knowledge has always been recognized, even in the Indian school of *Navya-Nyaya*, by Plato's Socrates (especially the dialogues *Meno*, and *Theaetetus*), and by Confucius. But its importance in the actual running of society has been only recently begun to be appreciated by those who do formal work. The work we described above indicates how rich the domain of interest is here.

References

1. Arrow, K.: *Social Choice and Individual Values*, Wiley (1963)
2. Aumann, R.: Agreeing to disagree, *Annals of Statistics*, **4** (1976) 1236-1239.
3. Brams, S., Taylor, A.: *The Win-Win Solution: guaranteeing fair shares to everybody*, Norton 1999.

4. Fagin, R., Halpern, J., Moses, Y., Vardi, M.: *Reasoning about Knowledge*, MIT Press 1995.
5. de Finetti, B.: Foresight: its Logical Laws, Its Subjective Sources, (translation of his 1937 article in French) in H. E. Kyburg H. E. Smokler (eds), *Studies in Subjective Probability*, New York: Wiley, 1964.
6. Geanakoplos, J., Polemarchakis, H.: We can't disagree forever, *J. Economic Theory*.
7. Hintikka, J.: *Knowledge Belief: an introduction to the logic of the two notions*, Cornell University press, 1962.
8. Lewis, D.: *Convention, a Philosophical Study*, Harvard U. Press, 1969.
9. von Neumann, J., Morgenstern, O.: *Theory of Games and Economic Behavior*, Princeton University Press, 1944 sec.ed. 1947
10. Pacuit, E., Parikh, R.: Reasoning about Communication Graphs, In *Interactive Logic: Selected Papers from the 7th Augustus de Morgan Workshop*, Johan van Benthem, Dov Gabbay and Benedikt Löwe (editors), Texts in Logic and Games, Volume 1, pgs. 135–157, 2007.
11. Pacuit, E., Parikh R.: Social Interaction, Knowledge, and Social Software, In *Interactive Computation: The New Paradigm*, Springer-Verlag 2006. Editors Dina Goldin, Scott Smolka and Peter Wegner.
12. Pacuit, E., Parikh, R., Cogan, E.: The Logic of Knowledge Based Obligation, *Knowledge, Rationality and Action*, a subjournal of *Synthese*, 149(2), 311 – 341, 2006.
13. Parikh, R.: The Logic of Games and its Applications, *Annals of Discrete Math.*, 24 (1985) 111-140.
14. Parikh, R.: Social Software, *Synthese*, **132**, Sep 2002, 187-211.
15. Parikh, R.: Levels of knowledge, games, and group action, in *Research in Economics*, **57**, (2003) 267-281.
16. Parikh R., Krasucki, P.: Communication, Consensus and Knowledge, *J. Economic Theory* **52** (1990) pp. 178-189.
17. Parikh, R., Parida, L., Pratt, V.: Sock Sorting, appeared in a volume dedicated to Johan van Benthem, University of Amsterdam, August 99, reprinted in *Logic J. of IGPL*, vol 9 (2001).
18. Parikh, R., Ramanujam, R.: 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, pp. 256-268.
19. Parikh, R., Ramanujam, R.: A knowledge based semantics of messages, in *J. Logic, Language, and Information*, **12**, pp. 453 - 467, 2003.
20. The Plame Affair, from the *Wikipedia*, http://en.wikipedia.org/wiki/Plame_affair
21. Ramsey, F. P.: Truth and probability, in *The Foundations of Mathematics and other Logical Essays*, (1926)
22. Savage, L.: *The Foundations of Statistics*, Wiley 1954.
23. Shoup, D.: Gone Parkin', Op-Ed page, *The New York Times*, March 29, 2007.
24. Turing, A.M.: On Computable Numbers, with an Application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society*, 2 42: 23065.