Social Interaction, Knowledge, and Social Software

Eric Pacuit[1] and Rohit Parikh[2]

[1] ILLC, Amsterdam, The Netherlands
[2] Brooklyn College and CUNY Graduate Center, New York, NY, USA

1 Introduction

In [31] a theory of *human computation*, analogous to Turing's theory of *machine computation*, is discussed. The issue there is whether there might be an analogue to Church's thesis in this human domain. Examples of human algorithms discussed include the making of scrambled eggs. By comparison, Lynn Stein in this volume discusses the making of a peanut butter and jelly sandwich. Neither she nor us in this volume have any concern with Church's thesis as such, although that might prove to be a fascinating topic for a future paper. Rather the issue here is *interaction*, which occurs most naturally in multiagent algorithms, unlike the making of scrambled eggs or peanut butter sandwiches where one agent normally suffices.[1] Such multiagent algorithms, examples of which are building a house, or playing bridge, are examples of what we shall call *social software* after [32]. In that paper, one of us asked "Is it possible to create a theory of how social procedures work with a view to creating better ones and ensuring the correctness of the ones we do have?" The present chapter will survey some of the logical and mathematical tools that have been developed over the years that may help address this question.

Social procedures occur at two levels. One is the purely personal level where an individual is able to perform some complex action *because* social structures have been set up to enable such an action. Taking a train (which requires a system) or even a bath (where the city must supply not only the water but also a system of pipes to carry it) are examples of such situations where an individual is doing something simple or complex which is enabled by existing social structures. Procedures which are *truly* social are those which require more than one individual even in their execution. A piano duet is a simple example, but holding an election or passing a bill through the Senate are more complex ones. Computer programs, whether sequential or distributed, have logical and algorithmic properties which can be analyzed by means of

[1] However, as the adage goes, it does take many cooks to *spoil* the broth!.

appropriate logics of programs. Similarly, these social procedures also have logical properties which can be analyzed by means of the appropriate logical tools, augmented by tools from game theory, perhaps even from psychology.

There are several ways to compare social software with distributed computing. In both cases the issue of knowledge arises. When several processes, whether human or computer, are taking part in a common procedure, then they need to know *enough* of what others are doing so as to be able to do their part when the time comes. Indeed, Halpern and Moses' fundamental paper on common knowledge was written in the context of distributed computing, although *other* authors like Aumann (game theory, see [2, 3]) and Lewis (social agreement, see [17]) had a different setting. Thus knowledge matters and we shall give a quick survey of current formal theories of knowledge.

However, unless the agents have the same goal, or at least compatible goals, there may be some element of strategizing where each agent tries to maximize its own benefit (sometimes represented as *utility*) while keeping in mind what other agents are apt to do. This makes game theory relevant.

In the context of social programming where an overarching social agent (say, a government) is trying to make agents act in a socially beneficial way, the social agent will still need to take into account the fact that while its own goal is *social welfare*, the goal of the individual agent is his own *personal welfare*. Thus agents have to be guided to act in beneficial ways. A simple example of this is the system of library fines to ensure that borrowers do not keep books too long and prevent other borrowers from having access to them.

Finally, agents may sometimes act in concert with other agents, i.e., form coalitions. There is an extensive theory of co-operative games but our primary purpose here will be to give a brief account of the logical theory of coalitions due to Marc Pauly.

Thus what we hope to do in this chapter is to survey some of these logical and analytical tools and indicate a few applications.

These tools are:

1. Logic of knowledge
2. Logic of games
3. Game theory and economic design

In the following sections we shall give brief descriptions of these three tools and then indicate some applications. We assume that the reader has some mild acquaintance with game theory (although we shall not actually use very much), and [16] is a good reference for that field. Moore [18] gives a survey of economic design. The sections are reasonably independent and the applications depend mainly on reasoning about knowledge.

2 Models of Knowledge and Belief

Formal models of knowledge and beliefs have been discussed by a diverse list of communities, including computer scientists ([7, 42, 27]), economists ([5, 2, 4]) and philosophers ([21, 11]). In this section we provide a brief overview of some of the models found in the computer science and game theory literature.

2.1 Epistemic Logic

Starting with Hintikka's *Knowledge and Belief* [21] there has been a lot of research on the use of logic to formalize the uncertainty faced by a group of agents. A detailed discussion of epistemic and modal logic and its applications in computer science can be found in the textbooks [7, 27].

The main idea of epistemic logic is to extend the language of propositional logic with symbols (K_i) that are used to formalize the statement "agent i knows ϕ" where ϕ is any formula. For example, the formula $K_i\phi \rightarrow \phi$ represents the widely accepted principle that agents can only know true propositions, i.e., if i knows ϕ, then ϕ must be true.

Formally, if At is a set of atomic propositions, then the language of multi-agent epistemic logic $\mathcal{L}_n^K(\mathsf{At})$ (or $\mathcal{LK}$ if At, n are understood from the context) has the following syntactic form:

$$\phi := A \mid \neg\phi \mid \phi \wedge \psi \mid K_i\phi$$

where $A \in \mathsf{At}$. We assume that the boolean connectives $\vee, \rightarrow, \leftrightarrow$ are defined as usual. The formula $L_i\phi$, defined as $\neg K_i \neg\phi$, is the dual of $K_i\phi$. Given that the intended meaning of the formula $K_i\phi$ is "agent i knows ϕ", $L_i\phi$ can be read as "ϕ is epistemically possible for agent i". There are a number of principles about knowledge—listed below—expressible in the language of epistemic logic that have been widely discussed by many different communities. Since our focus is on social software and not on epistemic or modal logic, we shall simply assume those schemes which correspond to the most widely prevalent understanding of the formal properties of knowledge. When more restricted properties of knowledge are entertained, *negative introspection* is the first axiom to be dropped. Let $\phi, \psi \in \mathcal{LK}$ be arbitrary formulas.

K	$K_i(\phi \rightarrow \psi) \rightarrow (K_i\phi \rightarrow K_i\psi)$	*Kripke's axiom*
T	$K_i\phi \rightarrow \phi$	*Truth*
4	$K_i\phi \rightarrow K_iK_i\phi$	*Positive introspection*
5	$\neg K_i\phi \rightarrow K_i\neg K_i\phi$	*Negative introspection*
D	$\neg K_i\bot$	*Consistency*

Note that D is a consequence of T.

We now turn to the semantics of epistemic logic. The main idea is that a formula $K_i\phi$ is true provided that ϕ is true in all situations that i considers possible. This definition was first put forward by Leibniz and is discussed in detail by Hintikka [21]. This intuition can be formalized using a Kripke structure.

Definition 1. *A* **Kripke model** *is a triple* $\langle W, \{R_i\}_{i\in\mathcal{A}}, V\rangle$ *where* W *is a nonempty set, for each* $i \in \mathcal{A}$*,* $R_i \subseteq W \times W$*, and* $V : \mathsf{At} \to 2^W$ *is a valuation function.*

In order to make sure that the axiom schemes $K, T, 4, 5, D$ hold, the relations R_i must all be equivalence relations. Elements $w \in W$ are called states, or worlds. We write wR_iv if $(w, v) \in R_i$. The relation R_i represents the uncertainty that agent i has about the "actual situation". In other words, if wR_iv and the actual situation is w, then for all agent i knows, the situation may be v. Notice that R_i represents the uncertainty each agent has about the actual situation and the agents' uncertainty about how the other agents view the situation, but it does not settle which basic facts are true at which states. For this, we need the valuation function V, where $w \in V(A)$ is interpreted as A *is true at state* w. We write $\mathcal{M}, w \models \phi$ to mean that ϕ is true at state w in $\mathcal{M}$. Truth is defined recursively as follows. Let $\mathcal{M} = \langle W, \{R_i\}_{i\in\mathcal{A}}, V\rangle$ be a model and $w \in W$ any state.

1. $\mathcal{M}, w \models A$ if $A \in V(s)$
2. $\mathcal{M}, w \models \phi \wedge \psi$ if $\mathcal{M}, w \models \phi$ and $\mathcal{M}, w \models \psi$
3. $\mathcal{M}, w \models \neg\phi$ if $\mathcal{M}, w \not\models \phi$
4. $\mathcal{M}, w \models K_i\phi$ if for each $v \in W$, if wR_iv, then $\mathcal{M}, w \models \phi$.

If the model $\mathcal{M}$ is understood we may write $w \models \phi$. If $\mathcal{M}, w \models \phi$ for all states $w \in W$, then we say that ϕ is **valid in** $\mathcal{M}$ and write $\mathcal{M} \models \phi$. Note that principle **4** is justified by the fact that i can only know ϕ if ϕ is true in every state where, for all i knows, he might be.

Common knowledge can be defined via the "everyone knows" operator. Let $E\phi = K_1\phi \wedge K_2\phi ... \wedge K_n\phi$, where $\mathcal{A} = \{1, ..., n\}$ is the set of agents. Thus $E\phi$ says that all n agents know ϕ. Then ϕ is "common knowledge" is expressed by the infinite conjunction $\phi \wedge E\phi \wedge E^2\phi \wedge ...$ For a more detailed discussion about reasoning about common knowledge see [15, 7]. See [17, 6] for a philosophical discussion of common knowledge.

2.2 Aumann Structures

One of the first attempts to formalize knowledge in economic situations is by Aumann [2]. As in the previous section, let W be a set of worlds, or states. In this section we reason semantically. Let S be the set of all states of nature. A state of nature is a complete description of the exogenous parameters (i.e., facts about the physical world) that do not depend on the agents' uncertainties.

In the previous section we defined an object language that could express statements of the form "agent i knows ϕ", and interpreted these formulas in a Kripke model. In this section we have no such object language. Reasoning about agents is done purely semantically. Thus we are making essential use of the fact that we can identify a proposition with the set of worlds in which it is true. Intuitively, we say that a set $E \subseteq W$, called an **event**, is true at state w if $w \in E$.

In [2], Aumann represents the uncertainty of each agent about the actual state of affairs by a partition over the set of states. Formally, for each agent $i \in \mathcal{A}$, there is a partition $\mathcal{P}_i$ over the set W. (A partition of W is a pairwise disjoint collection of subsets of W whose union is all of W.) Elements of $\mathcal{P}_i$ are called **cells**, and for $w \in W$, let $\mathcal{P}_i(w)$ denote the cell of $\mathcal{P}_i$ containing w. Putting everything together,

Definition 2. *An* **Aumann model based on S** *is a triple* $\langle W, \{P_i\}_{i\in\mathcal{A}}, \sigma\rangle$, *where* W *is a nonempty set, each* $\mathcal{P}_i$ *is a partition over* W *and* $\sigma : W \to S$.

So, σ is analogous to a valuation function, it assigns to each world a state of nature in which every ground fact (any fact *not about the uncertainty of the agents*) is either true or false. If $\sigma(w) = \sigma(w')$ then the two worlds w, w' will agree on all the facts, but the agents may have different knowledge in them. Elements of W are *richer* in information than the elements of S.

The event that agent i knows event E, denoted $\mathsf{K}_i E$, is defined to be

$$\mathsf{K}_i E = \{w \mid \mathcal{P}_i(w) \subseteq E\}$$

In other words, for each agent $i \in \mathcal{A}$, we define a set valued function $\mathsf{K}_i : 2^W \to 2^W$ using the above definition. It is not hard to show, given this definition and the fact that the $\mathcal{P}_i$ s are patitions, that for each $i \in \mathcal{A}$ and each $E \subseteq W$,

$$\begin{array}{ll}
E \subseteq F \Rightarrow \mathsf{K}_i(E) \subseteq \mathsf{K}_i(F) & \textit{Monotonicity} \\
\mathsf{K}_i(E \cap F) = \mathsf{K}_i(E) \cap \mathsf{K}_i(F) & \textit{Closure under intersection} \\
\mathsf{K}_i E \subseteq E & \textit{Truth} \\
\mathsf{K}_i(E) \subseteq \mathsf{K}_i(\mathsf{K}_i(E)) & \textit{Positive introspection} \\
\overline{\mathsf{K}_i(E)} \subseteq \mathsf{K}_i(\overline{\mathsf{K}_i(E)}) & \textit{Negative introspection} \\
\mathsf{K}_i(\emptyset) = \emptyset & \textit{Consistency.}
\end{array}$$

These are the analogues of the $K, T, 4, 5$ and D axiom schemes from the previous section. In fact, there is an obvious translation between Aumann structures and Kripke structures. In [14], Halpern formally compares the two frameworks pointing out similarities and important differences.

There is a more fine-grained model of uncertainty discussed in the game theory literature, usually called a Bayesian model. In a Bayesian model, the uncertainty of each agent is represented by probability functions over the set of worlds, and so we can express exactly *how* uncertain each agent is about the given situation. A detailed discussion and pointers to the relevant literature can be found in [5, 3].

Finally, a set E is a *common knowledge* set if $\mathsf{K}_i(E) = E$ for all i.[2] Event F is common knowledge at state w if there is a set E such that E is a

[2] Note that this definition makes heavy use of the richer state space W. Within E, agent i is not only aware of certain objective facts, she is also aware of some of the *knowledge* of other agents.

common knowledge set, and $w \in E \subseteq F$. Note that this definition of common knowledge is very transparent compared to the more syntactic one from the previous section.

2.3 History-Based Models

History based structures, also called *interpreted systems*, have been extensively discussed in the distributed computing literature (see [7] Chap. 4, 5 and 8 for a thorough discussion). This section will present the framework of Parikh and Ramanjam found in [35, 36]. In [36], Parikh and Ramanajam argue that this framework very naturally formalizes many social situations by providing a semantics of messages in which sophisticated notions such as Gricean implicature can be represented.

We begin by assuming the existence of a global discrete clock (whether the agents have access to this clock is another issue that will be discussed shortly). At each moment, some event takes place. Let E be a fixed set of events. As discussed in the previous section, it is natural to allow that different agents are aware of different events. To that end, assume for each agent $i \in \mathcal{A}$, a set $E_i \subseteq E$ of events "seen" by agent i. Before defining a history we need some notation: Given any set X (of events), X^* is the set of finite strings over X and X^ω the set of infinite strings over X. A **global history** is any sequence, or string, of events, i.e., an element of $E^* \cup E^\omega$. Let $h, h', \ldots$ range over E^* and $H, H', \ldots$ range over $E^* \cup E^\omega$. A **local history** for agent i is any element $h \in E_i^*$. Notice that local histories are always assumed to be finite.

Given two histories H and H', write $H \preceq H'$ to mean H is a *finite* prefix of H'. Let hH denote the concatenation of finite history h with possibly infinite history H. Let H_k denote the finite prefix of H of length k (given that H is infinite or of length $\geq k$). Given a set $\mathcal{H}$ of histories, define $\mathsf{FinPre}(\mathcal{H}) = \{h \mid h \in E^*, h \preceq H, \text{ and } H \in \mathcal{H}\}$. So $\mathsf{FinPre}(\mathcal{H})$ is the set of finite prefixes of elements of $\mathcal{H}$. A set $\mathcal{H} \subseteq E^* \cup E^\omega$ is called a **protocol**. Intuitively, the protocol is simply the set of possible histories that could arise in a particular situation. Following [36], little structure is placed on the set $\mathcal{H}$. I.e., the protocol can be *any* nonempty set of histories, provided only that if a history H is in the protocol $\mathcal{H}$, then so is any prefix of H. Notice that this notion of a protocol differs from standard usage of the term protocol which is taken to mean a procedure executed by a group of agents. Certainly any procedure will generate a set of histories, but not every set of histories can be generated by some procedure. Therefore, this definition of protocol is more general than the standard definition. It *is* useful as [36] use it to interpret even notions like Gricean implicature.

Given a particular finite global history H and an agent i, i will only "see" the events in H that are from E_i. This leads to a natural definition of agent uncertainty.

Definition 3. *For each $i \in \mathcal{A}$ define $\lambda_i : \mathsf{FinPre}(E^\omega) \to E_i^*$ to be the* **local view function** *of agent i.*

In systems in which agents cannot access a global clock. $\lambda_i(H)$ is obtained by mapping each event in E_i to itself and all *other* events to the empty string. Thus if $\lambda_i(H) = h$ for some finite history H, and event $e \in E_i$, which is visible to i, takes place next, then $\lambda_i(He) = he$, otherwise $\lambda_i(He) = h$. Let H and H' be two global histories in some protocol $\mathcal{H}$. We write $H \sim_{i,t} H'$ if according to agent i, H is "equivalent" to H' at time t, i.e., $\lambda_i(H_t) = \lambda_i(H'_t)$. It is easy to see that for each time $t \in \mathbb{N}$, $\sim_{i,t}$ is an equivalence relation.

Definition 4. *Given a history based multiagent frame for a set of agents $\mathcal{A}$ and events E, $\mathcal{F}_H = \langle \mathcal{H}, E_1, \ldots, E_n \rangle$, a* **history based model** *is a tuple $\langle \mathcal{H}, \lambda_1, \ldots, \lambda_n, V \rangle$, where each λ_i is a local view function and $V : \mathsf{FinPre}(\mathcal{H}) \to 2^{\Phi_0}$ is a valuation function.*

Finally, a few comments about whether agents have access to the global clock. We say that a history based frame $\mathcal{F}_H$ is **synchronous** if all agents have access to the global clock. Formally this is achieved by assuming a special event $c \in E$ with $c \in E_i$ for each $i \in \mathcal{A}$. This event represents a clock tick. In synchronous history based models, the local view function maps each event seen by agent i in some finite history H to itself, and all other events to the clock tick c. Notice that in such a case, for any finite global history H and local view function λ_i, the length of $\lambda_i(H)$ and the length of H are always equal.

Given these tree-like structures, it is natural to define a language in which we can express both knowledge-theoretic and temporal facts. Formally, we add a unary modal operator $\bigcirc$ and a binary modal operator U to the language $\mathcal{LK}$. Denote this language by $\mathcal{L}_n^{KT}$. $\bigcirc\phi$ is intended to mean that ϕ *is true after the next event* and $\phi U \psi$ is intended to mean that ϕ *is true until* ψ *becomes true*. Other well known temporal operators can be defined. Details can be found in [36] and [13, 7].

Truth is defined at finite histories. Thus, for $H \in \mathcal{H}$, $H, t \models \phi$ is intended to mean that in history H at time t, ϕ is true. Boolean connectives and atomic propositions are obvious.

1. $H, t \models \bigcirc\phi$ iff $H, t+1 \models \phi$
2. $H, t \models \phi U \psi$ iff there exists $m \geq t$ such that $H, m \models \psi$ and for all l such that $t < l < m$, $H, l \models \phi$
3. $H, t \models K_i\phi$ iff for all $H' \in \mathcal{H}$ such that $H \sim_{i,t} H'$, $H', t \models \phi$.

In the above definition of truth of K_i formulas (item 3 above), it is assumed that the agents all share a global clock. This assumption is made in order to simplify the presentation. A sound and complete axiomatization for knowledge and time under various assumptions can be found in [13], using a slightly different framework.

3 Logic of Games

The logic of games [33] is an offshoot of propositional dynamic logic or PDL. PDL was invented by Fischer and Ladner [8] following Pratt's work on first order dynamic logic.

In dynamic logic a program is thought of as running in a state space, and a program α is thought of as starting in some state s and arriving at some state t if and when it finishes. The program need not be deterministic so that starting with the same s it might instead arrive at some t'. This allows us to see α as a binary relation $R_\alpha = \{(s,t) | \alpha \text{ can go from } s \text{ to } t\}$. This converts α into a modality and allows us to define the constructs $[\alpha]$ and $\langle\alpha\rangle$, which are the program theoretic versions of the modal operators box and diamond. The formula $\langle\alpha\rangle A$ holds at state s if there is *some* run of the program α starting at s which results in a state t which satisfies A. $[\alpha]A$ holds if *every* terminating run does so.

However, our interest here is in *games* which can no longer be represented as binary relations, instead the semantics is more like the Scott–Montague semantics for modal logic in which Kripke's axiom **K** is no longer valid. The reason roughly is this. If α is a *program* and $\langle\alpha\rangle(A \vee B)$ holds then $\langle\alpha\rangle A$ or $\langle\alpha\rangle B$ must hold. For if there is an α-computation which results in $A \vee B$ then there must be one which results in A or one which results in B. ($\langle\alpha\rangle(A \vee B) \rightarrow \langle\alpha\rangle(A) \vee \langle\alpha\rangle(B)$ is an axiom equivalent to Kripke's **K**). But this need not hold with a game. It may well be that one player, say I, has a winning strategy to achieve $A \vee B$ in the game α without having a winning strategy to achieve either A reliably or B reliably. For instance a game of chess may reach a point where Black can ensure a checkmate in three moves, but it is White's moves which decide whether that checkmate is by queen or by rook—Black cannot ensure a checkmate by queen nor a checkmate by rook. Thus game logic is a non-normal (non-**K**) logic corresponding to PDL.

3.1 Syntax and Semantics

We have a finite supply $g_1, \ldots, g_n$ of atomic games and a finite supply $P_1, \ldots, P_m$ of atomic formulae. Then we define games α and formulae A by induction.

1. Each P_i is a formula.
2. If A and B are formulae, then so are $A \vee B$, $\neg A$.
3. If A is a formula and α is a game, then $(\alpha)A$ is a formula.
4. Each g_i is a game.
5. If α and β are games, then so are $\alpha;\beta$ (or simply $\alpha\beta$), $\alpha \vee \beta$, $\langle\alpha^*\rangle$, and α^d. Here α^d is the dual of α.
6. If A is a formula then $\langle A\rangle$ is a game.

We shall write $\alpha \wedge \beta$, $[\alpha^*]$ and $[A]$ respectively for the duals of $\alpha \vee \beta$, $\langle\alpha^*\rangle$ and $\langle A\rangle$. If confusion will not result then we shall write αA for $(\alpha)A$. For example, $\langle g_i^*\rangle A$ instead of $(\langle g_i^*\rangle)A$.

Intuitively, the games can be explained as follows. $\alpha;\beta$ is the game: play α and then β. The game $\alpha \vee \beta$ is: player I has the first move, she decides whether α or β is to be played, and then the chosen game is played. The game $\alpha \wedge \beta$ is similar except that player II makes the decision. In $\langle\alpha^*\rangle$, the game α is played repeatedly (perhaps zero times) until player I decides to stop. She need not declare in advance how many times is α to be played, but she is *required* to eventually stop, and player II may use this fact as part of his strategy. Player I may not stop in the *middle* of some play of α. Similarly with $[\alpha^*]$ and player II. In α^d, the two players interchange roles. Finally, with $\langle A\rangle$, the formula A is evaluated. If A is false, then I loses, otherwise we go on. (Thus $\langle A\rangle B$ is equivalent to $A \wedge B$.) Similarly with $[A]$ and II. The formula $(\alpha)A$ means that player I has a winning strategy to play game α in such a way that formula A is true if and when the game ends (or if the game does not end, the fault for that lies with II).

Formally, a model of game logic consists of a set W of worlds; for each atomic P a subset $\pi(P)$ of W; and for each primitive game g a subset $\rho(g)$ of $W \times P(W)$, where $P(W)$ is the power set of W. $\rho(g)$ must satisfy the monotonicity condition: if $(s,X) \in \rho(g)$ and $X \subseteq Y$, then $(s,Y) \in \rho(g)$. For clearly if an agent can play the game so as to be sure to be in X at the end, then the agent can also ensure Y by simply ensuring X. We shall find it convenient to think of $\rho(g)$ as an operator from $P(W)$ to itself, given by the formula

$$\rho(g)(X) = \{s|(s,X) \in \rho(g)\}$$

It is then monotonic in X. We define $\pi(A)$ and $\rho(\alpha)$ for more complex formulae and games as follows:

1. $\pi(A \vee B) = \pi(A) \cup \pi(B)$
2. $\pi(\neg A) = W - \pi(A)$
3. $\pi((\alpha)A) = \{s|(s,\pi(A)) \in \rho(\alpha)\} = \rho(\alpha)(\pi(A))$
4. $\rho(\alpha;\beta)(X) = \rho(\alpha)(\rho(\beta)(X))$
5. $\rho(\alpha \vee \beta)(X) = \rho(\alpha)(X) \cup \rho(\beta)(X)$
6. $\rho(\langle\alpha^*\rangle)(X) = \mu Y(X \subseteq Y \wedge \rho(\alpha)(Y) \subseteq Y)$
7. $\rho(\alpha^d)(X) = W - \rho(\alpha)(W - X)$
8. $\rho(\langle A\rangle)(X) = \pi(A) \cap X$.

It is easily checked that $\rho(\alpha \wedge \beta)(X) = \rho(\alpha)(X) \cap \rho(\beta)(X)$, $\rho([A])(X) = (W - \pi(A)) \cup X$, and $\rho([\alpha^*])(X) = \nu Y((Y \subseteq X) \wedge (Y \subseteq \rho(\alpha)(Y))$ where νY means "the largest Y such that". This is easily seen by noticing that $\rho([\alpha^*])(X) = W - \rho(\langle\alpha^*\rangle)(W - X) = W -$ the smallest Z such that $(W - X) \subseteq Z$ and $\rho(\alpha)(Z) \subseteq Z$.

We shall have occasion to use both ways of thinking of ρ, as a map from $P(W)$ to itself, also as a subset of $W \times P(W)$. In particular we shall need the (easily checked) fact that $(s,X) \in \rho(\beta;\gamma)$ iff there is a Y such that $(s,Y) \in$

$\rho(\beta)$ and for all $t \in Y$, $(t, X) \in \rho(\gamma)$. Similarly, $(s, X) \in \rho(\beta \vee \gamma)$ iff $(s, X) \in \rho(\beta)$ or $(s, X) \in \rho(\gamma)$.

So far we have made no connection with PDL. However, given a language of PDL we can associate with it a game logic where to each program a_i of PDL we associate two games $\langle a_i \rangle$ and $[a_i]$. We take $\rho(\langle a \rangle)(X) = \{s : \exists t(s,t) \in R_a$ and $t \in X\}$ and $\rho([a])(X) = \{s : \forall t(s,t) \in R_a$ implies $t \in X\}$ and the formulae of PDL can be translated easily into those of game logic. Note that if the program a is to be run and player I wants to have A true after, then if she runs a, only $\langle a \rangle A$ needs to be true. However, if player II is going to run the program a then $[a]A$ needs to be true for I to win in any case. Note that if there are no a-computations beginning at the state s, then player II is unable to move, $[a]A$ is true and player I wins. In other words, unlike the situation in chess, a situation where a player is unable to move is regarded as a loss for that player in both PDL and game logic.

However, game logic is more expressive than PDL. The formula $\langle [b^*] \rangle \texttt{false}$ of game logic says that *there is no infinite computation of the program* b, a notion that cannot be expressed in PDL.

Finally, let us show how well-foundedness can be defined in game logic. Given a linear ordering R over a set W, consider the model of game logic where g denotes $[a]$ and R_a is the inverse relation of R. Then R is well-founded over W iff the formula $\langle g^* \rangle \texttt{false}$ is true. Player I cannot terminate the game without losing, but she is required to terminate the game sometime. The only way she can win is to keep saying to player II, *keep playing!*, and hope that player II will sooner or later be deadlocked. (The subgame $[a]$ of $\langle [a]^* \rangle$ is a game where player II moves, and in the main game $\langle [a]^* \rangle$, player I is responsible for deciding how many times is $[a]$ played.) Thus I wins iff there are no infinite descending sequences of R on W.

However, despite its power, game logic can be translated into μ-calculus of [19] and by the decision procedure of [20], is decidable. An elementary decision procedure for dual-free game logic exists as does a completeness result, whose axiomatization is given below.

3.2 Completeness

The following axioms and rules are complete for the "dual-free" part of game logic.

The axioms of game logic

1. All tautologies
2. $(\alpha; \beta)A \Leftrightarrow (\alpha)(\beta)A$
3. $(\alpha \vee \beta)A \Leftrightarrow (\alpha)A \vee (\beta)A$
4. $(\langle \alpha^* \rangle)A \Leftrightarrow A \vee (\alpha)(\langle \alpha^* \rangle)A$
5. $(\langle A \rangle)B \Leftrightarrow A \wedge B$

Rules of inference

1. Modus ponens

$$\frac{A \quad A \Rightarrow B}{B}$$

2. Monotonicity

$$\frac{A \Rightarrow B}{(\alpha)A \Rightarrow (\alpha)B}$$

3. Bar induction

$$\frac{(\alpha)A \Rightarrow A}{(\langle\alpha^*\rangle)A \Rightarrow A}$$

The soundness of these axioms and rules is quite straightforward. The completeness proof given in [33].

The completeness problem for game logic *with* dual has now been open for about 20 years.

4 Coalitional Logic

In his dissertation [40], Marc Pauly extended game logic to a logic for reasoning about coalitional powers in games. This section will describe his basic framework. The interested reader is referred to [40, 39] for a more detailed discussion.

In game logic, the formula $[\alpha]\phi$ is intended to mean that player II has winning strategy in the determined, zero-sum game α. The intuition driving the semantics for game logic is that when $w\rho_\alpha X$ holds, player I (alone) can force the outcome of the game α to end in one of the states in X. Pauly drops the assumption of determinacy of the games, weakening the power of the individual players. In Pauly's semantics, typically a *coalition* of agents is needed for the outcome to end in some state in a set X.

The first step is the introduce a language that can express facts about coalitions of players. Given a finite set of agents $\mathcal{A}$, the language of coalitional logic has the following syntactic form

$$\phi \ := \ A \mid \neg\phi \mid \phi \vee \psi \mid [C]\phi$$

where $A \in \mathsf{At}$ is an atomic proposition and $C \subseteq \mathcal{A}$. The other boolean connectives are defined as usual. The intended interpretation of $[C]\phi$ is that the group of agents in C have a joint strategy to ensure that ϕ is true.

The semantics is essentially a Scott–Montague neighborhood model with a neighborhood function for each subset of agents. Let W be a set of states. An **effectivity function** is a map

$$E : (2^{\mathcal{A}} \times W) \to 2^{2^W}$$

We write wE_CX if $X \in E(C, w)$. The intended interpretation of wE_CX is that in state w, the agents in C have a joint strategy to bring about one of the states in X. An effectivity function is **playable** iff for all $w \in W$,

1. For all $C \subseteq \mathcal{A}$, $\emptyset \notin E(C, w)$
2. For all $C \subseteq \mathcal{A}$ $W \in E(C, w)$
3. E is **$\mathcal{A}$-maximal**, i.e., for all $X \subseteq W$, if $X \in E(\mathcal{A}, w)$ then $\overline{X} \notin E(\emptyset, w)$
4. E is **outcome-monotonic**, i.e., for all $X \subseteq X' \subseteq W$, $w \in W$, and $C \subseteq \mathcal{A}$, if $X \in E(C, w)$ then $X' \in E(C, w)$
5. E is **superadditive**, i.e., for all subsets X_1, X_2 of W and sets of agents C_1, C_2 such that $C_1 \cap C_2 = \emptyset$ and $X_1 \in E(C_1, w)$ and $X_2 \in E(C_2, w)$, we have $X_1 \cap X_2 \in E(C_1 \cup C_2)$.

Pauly [40] shows that these conditions are exactly the conditions needed to formalize the intuitive interpretation of the effectivity functions. Given any strategic game G, we can define an effectivity function generated by G. Essentially, we say that a set X is in $E_G(C)$ for some set $C \subseteq \mathcal{A}$ iff there is a strategy that the agents in C can play such that for any strategy that the other players follow, the outcome will be some element of X. Pauly showed that the above conditions charactize all effectivity functions generated by some game.

Theorem 1 (Pauly [40]). *An effectivity function E is playable iff it is the effectivity function E_G of some strategic game G.*

We can now formally define a coalitional model.

Definition 5. *A* **coalitional model** *is a tuple $\langle W, E, V\rangle$ where W is a nonempty set of states, E is a playable effectivity function, and $V : \mathsf{At} \to \mathcal{P}(S)$ is a valuation function.*

Given such a model, truth is defined as follows

$$\begin{array}{ll}
\mathcal{M}, w \models A & \text{iff } A \in \mathsf{At} \text{ and } w \in V(A) \\
\mathcal{M}, w \models \neg\phi & \text{iff } \mathcal{M}, s \not\models \phi \\
\mathcal{M}, w \models \phi \vee \psi & \text{iff } \mathcal{M}, s \models \phi \text{ or } \mathcal{M}, w \models \psi \\
\mathcal{M}, w \models [C]\phi & \text{iff } wE_C\phi^{\mathcal{M}}
\end{array}$$

where $\phi^{\mathcal{M}} = \{w \in W \mid \mathcal{M}, w \models \phi\}$. Pauly shows [40] that the following axiom system is sound and complete for the class of coalitional models.

$$\begin{array}{ll}
(\bot) & \neg[C]\bot \\
(\top) & [C]\top \\
(N) & \neg[\varnothing]\neg\phi \to [N]\phi \\
(M) & [C](\phi \wedge \psi) \to [C]\psi \\
(S) & ([C_1]\phi_1 \wedge [C_2]\phi_2) \to [C_1 \cup C_2](\phi_1 \wedge \phi_2)
\end{array}$$

provided $C_1 \cap C_2 = \varnothing$. We also assume modus ponens and that from $\phi \leftrightarrow \psi$, we can infer $[C]\phi \leftrightarrow [C]\psi$.

5 Some Applications

Our primary purpose in this survey has been to give a survey of tools used in studying social software. However, we now proceed to give some examples of applications. The first two examples are *light.*

5.1 A Knowledge Interaction

Suppose that Bob is giving a seminar and would like Ann to attend his talk; however, he only wants Ann to attend if she is interested in the subject of his talk, not because she is just being polite.

Why can't Bob just *tell* Ann about his talk?

We suggest that Bob would like to satisfy three conditions.

1. $K_a(S)$ (Ann knows S, where S stands for the proposition that Bob is giving the seminar.)
2. $K_bK_a(S)$ (Bob knows that Ann knows S.)
3. $\neg K_aK_bK_a(S)$ (Ann does *not* know that Bob knows that she knows S.)

Let us examine the three conditions. Clearly the first is necessary, for if Ann does not know about the seminar she cannot go, even if she wants to. The second, while not crucial, gives Bob peace of mind.

It is the last one which is interesting. Ann could have two reasons for going. She could go because she is interested in the talk. Or she could go to please Bob or out of fear that he will be offended if she does not go. If she knows that Bob knows that she knows, she will have to allow for an expectation on his part that she should go.

If Bob just *tells* her about the seminar, then common knowledge of S will be created, including the dreaded formula $K_aK_bK_a(S)$. So Bob cannot just *tell* her.

But he *can* ask a friend discreetly to tell her. Then he will be more confident that she will not feel pressured to come. This solves his problem of achieving the three conditions 1–3.

A similar example arises with a joke about a butler in a hotel who enters a room to clean it, and surprises a woman guest coming out of the bath. "Excuse me, sir, and he withdraws."

Why "*sir*"? Because she can reason that if he is mistaken about the gender, then he could not have seen her clearly, and there is no reason for her to be embarrassed—or to complain to the hotel. The butler very intelligently saves her from embarrassment by deliberately creating a false belief in her. (In other words $\neg K_gK_b(F)$ and even $B_g\neg K_b(F)$ where F stands for the fact that the guest is female, and B is the belief operator.)

Such issues will arise again in the section on knowledge based obligation.

It is generally accepted that what people do depends on what they believe, what they prefer, and what their options are. Their beliefs tell them what the options are and how they should be weighed. Thus if Bob has the option of meeting Jane for dinner or not, but does not know if she is pretty or ugly, then in a sense he knows what his options are, to meet her or not. But there is also a sense in which he does not know how to weigh the options. Now if he knows that Jane is ugly, he can safely have dinner with her without worrying that his own wife will be suspicious.

In the same way, in our earlier example, Ann does have the option of going to the seminar or not—once she knows about it. But how she weighs that option will depend on whether she knows that Bob knows that she knows.

5.2 The Two Horsemen and Letters of Recommendation

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 Mary, 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 she will give a prize of \$100. However, there is an interesting twist. She 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 Mary's face as she sees that she is having some amusement at no cost. Each horseman can make his horse go at any speed upto its maximum. But he has no reason to use the maximum. They try to go as slow as they can and so they end up in a stalemate with both horses going at 0 miles per hour. Let x, y be the speeds respetively at which Bill's horse and Joe's horse are going. Then [0,0] is a Nash equilibrium here.

However, along comes another passerby, let us call her Pam, 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! Joe's horse, ridden by Bill, comes first and Bill gets the \$100 as he should. The Nash equilibrium has shifted to [35,30].

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. Some 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, we know no obvious analogue to the strategem of exchanging horses. Certainly, if someone were to find such an analogue, it would revolutionize the whole process of writing letters of recommendation.

5.3 Banach–Knaster Cake Cutting Procedure

The following problem has often been mentioned in the literature. Some n people have to share a cake and do not have access to any measuring device. Moreover, they do not trust each other. Can they *still* divide the cake in a way which seems fair to all? The Banach–Knaster last diminisher procedure goes as follows.

Player 1 cuts out a piece p which she claims is a fair share for her. After that p is inspected by the other $n-1$ people. Anyone who thinks the piece too big may put something back into the main cake. After all $n-1$ have looked at it, one of two things must have happened. Either no one diminished p, in which case player 1 takes p and leaves to eat it. Or else one or more people *did* diminish p in which case the *last diminisher* takes the reduced p and leaves. In any case, the game is now down to $n-1$ people and can be repeated.

It is proved in [33] that this procedure is correct in the sense that each of the n players has a winning strategy to make sure that he gets his fair share. The technique used uses an n person (rather than two-person) version of game logic of Sect. 3.

5.4 Consensus

In 1979 Robert Aumann proved a spectacular result [1]. Suppose that two people A, B with the same prior probability distibution receive *different* information about some event E. It is then likely that their probabilities for E will diverge and that $p = p_A(E)$ could be different from $q = p_B(E)$. What Aumann showed was that if the values p and q are *common knowledge* then they must be equal. This result (somewhat extended) has the following curious consequence: suppose that A is planning to sell B a stock at a selling price s and B is plannning to buy. Assuming that they are both motivated by money and not, say by love or hate for the stock, the future price which A expects the stock to have is *less* than s and the future price which B expects the stock to have is *more* than s. But this fact is common knowledge as it is of course common knowledge that the sale is taking place. But this violates the theorem, the future prices *cannot* be different and the sale cannot take place! This is indeed a paradoxical result.

Aumann's result was extended by Bacharach, Cave, and Geanakoplos and Polmarchakis [10]. The last two showed that in Aumann's framework, if p, q

were not common knowledge they *could* be different, but that if the values $p_A(E)$ and $p_B(E)$ were repeatedly exchanged by A, B, and repeatedly revised, then the process of revision would eventually make them equal. A result by Parikh and Krasucki [34] extends the same phenomenon to n agents who communicate pairwise in a strongly connected graph. It is shown that personal values of probabilities and other strongly convex functions eventually become equal when people communicate in pairs, provided that no one is left out of the chain.

5.5 Logic of Communication Graphs

In [29], Pacuit and Parikh introduce a multimodal epistemic logic for reasoning about knowledge and communication. The language is a multiagent modal language with a communication modality. The formula $K_i\phi$ is interpreted as "according to i's current information, i knows ϕ", and $\Diamond\phi$ will be interpreted as "after some communications among the agents, ϕ becomes true". Thus for example, the formula

$$K_j\phi \rightarrow \Diamond K_i\phi$$

expresses that if agent j (currently) knows ϕ, then after some communication agent i can come to know ϕ. The following example illustrates the type of situations that the logic of communicationg graphs is intended to capture.

Consider the current situation with Bush and Porter Goss, the director of the CIA. If Bush wants some information from a particular CIA operative, say Bob, he must get this information through Goss. Suppose that ϕ is a formula representing the exact whereabouts of Bin Laden and that Bob, the CIA operative in charge of maintaining this information knows ϕ. In particular, $K_{\mathrm{Bob}}\phi$, but suppose that at the moment, Bush does not know the exact whereabouts of Bin Laden ($\neg K_{\mathrm{Bush}}\phi$). Presumably Bush *can* find out the exact whereabouts of Bin Laden ($\Diamond K_{\mathrm{Bush}}\phi$) by going through Goss, but of course, *we* cannot find out such information ($\neg\Diamond K_{\mathrm{e}}\phi \wedge \neg\Diamond K_{\mathrm{r}}\phi$) since we do not have the appropriate security clearance. Clearly, then, as a *prerequisite* for Bush learning ϕ, Goss will also have come to know ϕ. We can represent this situation by the following formula:

$$\neg K_{\mathrm{Bush}}\phi \wedge \Box(K_{\mathrm{Bush}}\phi \rightarrow K_{\mathrm{Goss}}\phi)$$

where $\Box$ is the dual of diamond. And this is because there is no direct link between Bush and Bob, only a chain going through Goss.

It is assumed that a set At of propositional variables are understood by (in the language of) all the agents, but only specific agents know their actual values at the start. Thus initially, each agent has some private information which can be shared through communication with the other agents. Now, if agents are restricted in whom they can communicate with, then this fact will restrict the knowledge they can acquire.

Let $\mathcal{A}$ be a set of agents. A **communication graph** is a directed graph $\mathcal{G}_{\mathcal{A}} = (\mathcal{A}, E)$ where $E \subseteq \mathcal{A} \times \mathcal{A}$. Intuitively $(i, j) \in E$ means that i can directly receive information from agent j, but *without* j knowing this fact. Thus an edge between i and j in the communication graph represents a one-sided relationship between i and j. Agent i has access to any piece of information that agent j knows. We have introduced this "one sidedness" restriction in order to simplify our semantics, but also because such situations of one sided learning occur naturally. A common situation that is helpful to keep in mind is accessing a website. We can think of agent j as creating a website in which everything he *currently* knows is available, and then if there is an edge between i and j then agent i can access this website without j being aware that the site is being accessed. Another important application of course is spying, where one person accesses another's information without the latter being aware that information is being leaked. Naturally j may have been able to access some other agent k's website and had updated some of her own information. Therefore, it is important to stress that when i accesses j's website, he is accessing j's current information which may include what another agent k knew initially.

The semantics combines ideas both from the subset models of [28] and the history based models of Parikh and Ramanajum (see [35, 36] and Sect. 2.3). The reader is refered to [29] for the details of the semantics. The satisfiability problem for the logic of communication graphs is shown to be decidable. Furthermore, as one may suspect, there is a connection between the structure of the communication graph and the set of valid formulas in a model (based on the communication graph). The following formula

$$\bigwedge_{l} K_j\phi \wedge \neg K_l\phi \rightarrow \Diamond(K_i\phi \wedge \neg K_l\phi)$$

where i, j are distinct agents, l ranges over agents distinct from these two and ϕ is a ground formula, states that it is possible for i to learn ϕ from j without any other l learning ϕ. Intuitively, this should be true if i has access to j's website without interference from anyone. It is shown in [29] that if there is an edge from i to j in a graph $\mathcal{G}$ then the above formula scheme is valid in the model based on $\mathcal{G}$.

5.6 Knowledge-Based Obligation

We start with the intuition that agents cannot be expected to perform actions, the need for which they are not aware of. In [30], Parikh, Pacuit and Cogan present a multiagent logic of knowledge, action and obligation. The semantics extends the history based models described in Sect. 2.3. In [30], various deontic dilemmas are described that illustrate the dependency of an agent's obligation on knowledge. For instance a doctor cannot be expected to treat a patient unless she is *aware* of the fact that he is sick, and this creates a secondary obligation on the patient or someone else to *inform* the doctor of his situation. In other words, many obligations are situation dependent, and are only relevant in the presence of the relevant information. This creates the notion of *knowledge-based obligation.*

Both the case of an absolute obligation (although dependent on information) as well as the notion of an obligation which may be over-ridden by more relevant information are considered. For instance a physician who is about to inject a patient with drug d may find out that the patient is allergic to d and that she should use d' instead. Dealing with the second kind of case requires a resort to nonmonotonic reasoning and the notion of *weak knowledge* which is stronger than plain belief, but weaker than absolute knowledge in that it can be over-ridden. Consider the following examples:

(a) Uma is a physician whose neighbor is ill. Uma does not know and has not been informed. Uma has no obligation (as yet) to treat the neighbor.
(b) Uma is a physician whose neighbor Sam is ill. The neighbor'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.
(c) 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. When there is a knowledge based obligation, but also the obligation to *have* the knowledge, then we have an obligation *simpliciter.*
(d) Uma has a patient with a certain condition C who is in the St. Gibson hospital mentioned above. There are two drugs d and d' which can be used for C, but d has a better track record. Uma is about to inject the patient with d, but unknown to Uma, the patient is allergic to d and for this patient d' should be used. Nurse Rebecca is aware of the patient's allergy and also that Uma is about to administer d. It is then Rebecca's obligation to inform Uma and to suggest that drug d' be used in this case.

In all the cases we mentioned above, the issue of an obligation arises. This obligation is circumstantial in the sense that in other circumstances, the obligation might not apply. Moreover, the circumstances may not be fully known. In such a situation, there may still be enough information about the circumstances to decide on the proper course of action. If Sam is ill, Uma needs to know that he is ill, and the nature of his symptoms, but not where Sam went to school.

Suppose that you want to formalize Uma's reasoning in the above examples, and *formally prove* that she is obliged to treat Sam in example (b). This has in fact been one of the goals of standard *deontic logic.* See [23, 22] and references therein for an uptodate discussion of deontic logic. Getting back to formalizing Uma's reasoning, one of the main points discussed above is that Uma's obligation arises only after she learns of her neighbor's illness. In other words, her obligation depends on her having the appropriate knowledge. In much of the deontic logic literature, an agent's knowledge is only informally represented or the discussion is focused on representing epistemic obligations, i.e., what an agent "ought to know", see [26] for a recent discussion. The logic in [30] is intended to capture the *dependency* of individual obligation on knowledge. The semantics extends the history based models described in Sect. 2.3 with PDL-style action modalities and a deontic operator. Refer to [30] for a detailed discussion of the semantics.

6 Conclusion

We end this paper with an amusing story about Mark Twain.

> 'There was a mystery,' said I. 'We were twins, and one day when we were two weeks old—that is, he was one week old and I was one week old—we got mixed up in the bathtub, and one of us drowned. We never could tell which. One of us had a strawberry birthmark on the back of his hand. There it is on my hand. This is the one that was drowned. There's no doubt about it.'
>
> 'Where's the mystery?' he said.
>
> 'Why, don't you see how stupid it was to bury the wrong twin?' I answered.
>
> (Mark Twain in a 1906 interview reported by the *New York Times*)

The *New York Times* reporter was not fast enough on his feet to hoist Twain on his own petard and ask what difference it made which twin was buried if people could not tell them apart (even after the drowning). But Twain's joke, like other deep jokes (by Groucho Marx or by the Sufi Mullah Nasruddin) leads into important issues like why we need names for people, why the government needs social security numbers, why identity theft is possible.

Who am I? is normally a question which typically a Zen Buddhist asks. But *Who are you?* is a question which others ask quite often. And this is because societal algorithms depend very much on *identity.* The bank does not want to allow others to withdraw funds from our accounts, or to allow us to withdraw funds from the accounts of others. Questions can be raised here at two levels. One level is why algorithms work only when identity is established. But a deeper level is what game theoretic reasons lie behind such algorithms in the first place. For instance in the play *Romeo and Juliet* when a Montressor has killed a Capulet, it is fine to kill another Montressor to revenge oneself. So the identity which matters here is not personal, but based on clan. There is a game between the two clans, where a threat to kill one member of a clan may be a deterrent on another. This is perhaps a foolish "algorithm", where one Montressor is killed instead of another, but favours are also often dealt out for similar reasons. These issues of the importance of (personal or tribal) identity to the correctness and relevance of games are deep and belong to another (future) paper. But we hasten to point out that they *are* urgent. When Sunni Arabs explode a bomb at a Shia mosque in Iraq, they may have nothing against the individual Shias praying at the mosque. They are *sending a message* to the *group*. If we want to solve such problems, we will surely need to go into the question of interactions where what matters is group identity and not personal identity.

References

1. R. Aumann. Agreeing to Disagree, *Annals of Statistics*, 4:1236 – 1239.
2. R. Aumann. Interactive Epistemology I: Knowledge, *International Journal of Game Theory*, 28:263–300, 1999.
3. R. Aumann. Interactive Epistemology II: Probability, *International Journal of Game Theory*, 28:301–314, 1999.
4. A. Brandenburger, Knowledge and Equilibrium in Games, *Journal of Economic Perspectives*, Vol. **6**, 1992, pp. 83–101.
5. G. Bonanno, and P. Battigalli. Recent results on belief, knowledge and the epistemic foundations of game theory, *Research in Economics 53*, 2 (June 1999), 149–225.
6. M. Chwe, *Rational Ritual : Culture, Coordination, and Common Knowledge*, Princeton University Press, 2001.
7. R. Fagin, J. Halpern, Y. Moses, and M. Vardi. *Reasoning about Knowledge*, The MIT Press, 1995.
8. M. J. Fischer and R. E. Ladner. Propositional dynamic logic of regular programs *J. Comput. Syst. Sci.*, 18(2), pp. 194–211.
9. R. W. Floyd. Assigning meanings to programs, *Proc. Symp. Appl. Math.*, Volume **19**, pp. 19–31.
10. J. Geanakoplos and H. Polemarchakis. We Can't Disagree Forever, *Journal of Economic Theory*, **28(1)**, 1982.
11. P. Gochet and P. Gribomont. Epistemic logic, In *The Handbook of History and Philosophy of Logic*, D. Gabbay and J. Woods, Eds., vol. 4. Elsevier, forthcoming.
12. D. Harel, D. Kozen and J. Tiuryn. *Dynamic Logic*, MIT Press, 2000.
13. J. Halpern, R. van der Meyden and M. Vardi. Complete Axiomatizations for Reasoning about Knowledge and Time, *SIAM Journal on Computing*, Vol **33**, No. 3, 2004, pp. 674–703.
14. J. Halpern. Set-theoretic completeness for epistemic and conditional logic, *Annals of Mathematics and Artificial Intelligence 26* 1999, pp. 1–27.
15. J. Halpern and Y. Moses, Knowledge and common knowledge in a distributed environment, *Journal of the ACM*, **37:3**, 1990, pp. 549-587.
16. M. Osborne and A. Rubinstein. *A Course in Game Theory*, MIT Press, 1994.
17. D. Lewis, *Convention: A Philosophical Study*, Harvard University Press, 1969.
18. J. Moore. Implementation in Environments with Complete Information, In J.J. Laffont, *Advances in Economic Theory: Proceedings of the Congress of the Econometeric Society*, Cambridge University Press, 1992.
19. D. Kozen. *Results on the propositional μ-calculus*, Proc 9th ICALP, Springer LNCS #140, 1982, pp. 348–359.
20. D. Kozen and R. Parikh, *A decision procedure for the propositional μ-calculus*, Proc. CMU Conf. on the Logic of Programs, Springer LNCS #164, pp. 313–325.

21. J. Hintikka. *Knowledge and Belief: An Introduction to the Logic of the Two Notions*, Cornell University Press, 1962.
22. R. Hilpinen. Deontic Logic, in *Blackwell Guide to Philosophical Logic*, Ed. Lou Goble, Blackwell, 2001, pp. 159–182.
23. J. Horty. *Agency and Deontic Logic*, Oxford 2001.
24. C. A. R. Hoare. An axiomatic basis for computer programming, *Comm. ACM* 12, pp. 576–580, 583.
25. B. Kooi. *Knowledge, chance and change* Ph.D. thesis, 2003.
26. A. Lomuscio and M. Sergot. Deontic interpreted systems, *Studia Logica*, **75**, 2003, pp. 63–92.
27. J.J. Meyer and W. van der Hoek. *Epistemic Logic for Computer Science and Artificial Intelligence* Cambridge Tracts in Theoretical Computer Science 41, Cambridge University Press, 1995.
28. L. Moss and R. Parikh. Topological Reasoning and the Logic of Knowledge, *TARK IV*, Ed. Y. Moses, Morgan Kaufmann, 1992.
29. E. Pacuit and R. Parikh. The Logic of Communication Graphs, *Proc. DALT 2004*, Joao Alexandre Leite, Andrea Omicini, Paolo Torroni, Pinar Yolum (Eds.), Revised Selected Papers, Springer LNCS #3476, 2005, pp. 256–269.
30. E. Pacuit, R. Parikh, and E. Cogan. The Logic of Knowledge Based Obligations, Presented at DALT 2004, forthcoming in *Knowledge, Rationality and Action* 2005.
31. R. Parikh Effectiveness, *the Philosophical Forum* XII, 1980, pp. 68–81.
32. R. Parikh. Social software, *Synthese*, pp. 187–211, September 2002.
33. R. Parikh. The Logic of Games and its Applications, *Annals of Discrete Math.*, 24, 1985, pp. 111–140.
34. R. Parikh and P. Krasucki. Communication, Consensus and Knowledge, *J. Economic Theory*, **52**, 1990, pp. 178–189.
35. R. Parikh and R. Ramanujam. Distributed Processing and the Logic of Knowledge, in *Logic of Programs*, Springer LNCS #193, June 1985, pp. 256–268.
36. R. Parikh and R. Ramanujam. A Knowledge based Semantics of Messages, *J. Logic, Language and Information*, **12**, 2003, pp. 453–467.
37. R. Parikh. Levels of Knowledge, Games, and Group Action, *Research in Economics*, vol 57, 2003, pp. 267–281.
38. R. Parikh. Logical omniscience, in *Logic and Computational Complexity* Ed. Leivant, Springer LNCS #960, 1995, pp. 22–29.
39. M. Pauly. A Logical Framework for Coalitional Effectivity in Dynamic Procedures, in *Bulletin of Economic Research*, **53(4)**, pp. 305–324.
40. M. Pauly. Logic for Social Software, Ph.D. Thesis, University of Amsterdam. ILLC Dissertation Series 2001-10, ISBN: 90-6196-510-1.
41. V. Pratt. Semantical considerations on Floyd-Hoare logic, In *Proc. 17th Symp. Found. Comput. Sci.* pp. 109–121. IEEE.
42. A. M. Zanaboni. *Reasoning about knowledge: Notes of Rohit Parikh's lectures.* Published in Italy: Cassa di Risparmio di Padova e Rovigo, June 1991. Based on lectures given at the 3rd International School for Computer Science Researchers, Acireale, June 1991.