UCLA Working Papers in Linguistics
Volume v, Article a: 1–9, yyyy

Joint Knowledge

Marcus Kracht

Joint knowledge of a group is the maximal knowledge that members of a group can attain only by talking to each other. I propose a formal approach and show how — at least in principle — group members can find out whether a proposition is jointly known.

1 The Problem

On the one hand, communication serves to distribute knowledge. On the other hand, knowledge is presupposed in communication. This is not a contradiction. The knowledge that is presupposed in communication is the so called *common knowledge*, while that which is communicated cannot be, by pragmatic principles. If φ is known by everyone, the Principle of Informativeness discourages utterance of φ. As Williamson (2000) argues, an assertion is licit only when what gets communicated is known to the speaker. This means that it should not be known to the hearer (as far as the speaker knows, that is). The effect of the communication is that it makes the assertion common knowledge, see Balbiani, Baltag, van Ditmarsch, Herzig, Hosi, and de Lima (2008). The knowledge that can be so attained is limited to what is known by all the speakers. This I call *joint knowledge*. This is the same as the "implicit knowledge" defined in Halpern (1987), but that term strikes me as unfortunate. A proposition is known jointly by a group if it follows from the union of all the propositions known individually. The aim of this paper is to investigate this notion.

2 Definitions

Let G be a set, the group of *agents*. For each $a \in G$, let K_a be the operator "a knows that". I take it that K_a satisfies the postulates of some modal logic, be it KT (Williamson (2000)), $\mathsf{S4}$ (Hintikka (1962)) or $\mathsf{S5}$ (Fagin, Halpern, Moses, and Vardi (1995)). All these conditions are equivalent to universal elementary conditions on Kripke-frames. I use the notation of propositional dynamic logic (PDL, see Goldblatt (1987)). So, K_a is based on a so-called "program" κ_a, $a \in G$, which gets interpreted as a relation between states, called here as usual *worlds*. We present the arguments assuming tacitly that K_a satisfies $\mathsf{S5}$, the relation associated with κ_a is an equivalence relation $E_a \subseteq W \times W$, but little hinges on that. Given $w \in W$, the w-alternatives for a are all b for which $a \; E_a \; b$. These are also called the epistemic alternatives for a

at w. What is known to a at w is what is true in all w-alternatives for a. Thus $K_a\varphi$ is tantamount to $[\kappa_a]\varphi$. The more alternatives w has, the less is known to a. It follows that knowledge *increases* when the E_a gets refined (so that the equivalence classes shrink). Now, for a group $H \subseteq G$ denote by "$C_H\varphi$" the fact that it is common knowledge for all $a \in H$ that φ. The standard definition is this. C_H is based on a program γ_H defined by

(1) $$\gamma_H := (\bigcup_{a\in H} \kappa_a)^*$$

This is to say that $C_H\varphi$ is nothing but $[\gamma_H]\varphi$. As we close the union (reflexively and) transitively, this is again an equivalence relation. Common knowledge satisfies again the postulates of S5. Notice that nothing less than the transitive closure suffices, and it has been argued that these steps of iteration are strictly required in pragmatics. (See also the problems of imperfect communication in Halpern (1987).)

The definition of joint knowledge is however much simpler.

Definition 1. Let U_a be the set of propositions known to a and let $H \subseteq G$ be nonempty. The **joint knowledge of** H, U_H, is the deductive closure of $\bigcup_{a\in H} U_a$. "$J_H\varphi$" is short for $\varphi \in U_H$.

J_H is based on the program ι_H, which is defined as follows.

(2) $$\iota_H := \bigcap_{a\in H} \kappa_a$$

Notice that we require that H is not empty. If you are desperate, let $\iota_\varnothing$ be the total relation on the frame. For a world w' to be a w-alternative according to what the members of H know jointly, it must be an alternative for every member of H, for everyone needs to agree on the alternatives to the world w. Since the intersection of equivalence relations is again an equivalence relation, J_H also satisfies S5. This generalises to the weaker logics KT and S4 as a consequence of the following observation.

Theorem 2. *Let P be a variable for binary relations, x_i variables over worlds. Let $\varphi = \varphi(P, \overline{x})$ be a second order formula relations of the following kind. It is made from formulae of the form $x_i\ P\ x_j$ using conjunction, disjunction, and restricted and unrestricted universal quantification (which have the form $(\forall x_i)(x_i\ P\ x_j \to \cdot)$ and $(\forall x_i)$, respectively). If R and S are relations on a set M satisfying φ, then also $R \cap S$ satisfies φ.*

Proof. Let φ^R (φ^S, $\varphi^{R\cap S}$) be the result of inserting R (S, $R\cap S$) for P in φ. By induction on the formulae we show that for every first-order valuation β sending variables to worlds, $\langle\mathfrak{M}, \beta\rangle \vDash \varphi^R$ and $\langle\mathfrak{M}, \beta\rangle \vDash \varphi^S$ implies $\langle\mathfrak{M}, \beta\rangle \vDash \varphi^{R\cap S}$. For the atoms, this is clear. If $w\ R\ v$ and $w\ S\ v$ then $w\ (R \cap S)\ v$. The inductive steps for conjunction and disjunction are straightforward. Suppose now that $\langle\mathfrak{M}, \beta\rangle \vDash (\forall y)(x\ R\ y \to \varphi^R)$ and $\langle\mathfrak{M}, \beta\rangle \vDash (\forall y)(x\ S\ y \to \varphi^S)$. Choose a w and let $\beta'(y) := w$ be a y-variant of β. If w is not a $(R\cap S)$-successor of $\beta(x)$, we trivially have $\langle\mathfrak{M}, \beta'\rangle \vDash (x\ Py \to \varphi)^{R\cap S}$, since this formula is nothing but $(x\ (R \cap S) y \to \varphi^{R\cap S})$. Thus, let us assume that $\beta(x)\ (R \cap S)\ w$.

Then $\beta(x)\,R\,w$ and so $\langle\mathfrak{M},\beta'\rangle \vDash \varphi^R$. By the same reasoning, $\langle\mathfrak{M},\beta'\rangle \vDash \varphi^S$. Hence $\langle\mathfrak{M},\beta'\rangle \vDash \varphi^{R\cap S}$, by inductive assumption, and so $\langle\mathfrak{M},\beta'\rangle \vDash (x\ P\ y \to \varphi)^{R\cap S}$ also in this case. β' was an arbitrary y-variant of β. Hence $\langle\mathfrak{M},\beta\rangle \vDash (\forall y)(x\ P y \to \varphi)^{R\cap S}$. Unrestricted quantification is similar. □

Notice how joint knowledge can be defined without an auxiliary notion (as the E_G operator, which codifies "everybody in the group knows", whose transitive closure is C_G). Reflexivity is $(\forall x)(x\,R\,x)$, symmetry $(\forall x)(\forall y)(x\,R\,y \to y\,R\,x)$ and transitivity is $(\forall x)(\forall y)(x\,R\,y \to (\forall z)(y\,R\,z \to x\,R\,z))$, and so all three conditions are of the form required by the theorem.

The axiomatisation of common knowledge proceeds by axiomatising the closure, which is already part of PDL. The intersection is not part of PDL, however. The extension of PDL with intersection is not straightforward, since intersection is not modally definable, see Passy and Tinchev (1991) for a discussion. Adding the axiom $\langle\alpha\cap\beta\rangle\varphi \to \langle\alpha\rangle\varphi \wedge \langle\beta\rangle\varphi$ is not enough (the converse implication is clearly false), and something much stronger needs to be added as well, for example nominals, for it simply encodes that $\alpha\cap\beta$ is contained in α and β, not that it is identical to them.

3 Communicating Knowledge

The main point of this paper is however not the axiomatisation of joint knowledge. The question is its role in communication. We refer here to the framework of Brandt and Kracht (2011) for communication in a network. A network consists of a set G of agents together with with a set $\mathscr{C} \subseteq \wp(G)$ of so-called *channels*. The communication structure of Brandt and Kracht (2011) further adds an addressing mechanism, whose role can be ignored here. A channel $C \in \mathscr{C}$ allows to transmit a message from one member of C to all other members. To make matters simple, we allow only the following kinds of messages to be sent: "?φ", the question whether φ is true, to which recipients may answer with "yes" (if they know that φ), "no" (if they know that $\neg\varphi$) or "don't know" (if they neither know that φ or that $\neg\varphi$); further, "!φ", the announcement that φ is true. To stay with the symmetrical flavour of Brandt and Kracht (2011), "!φ" must be followed by the acknowledgment "ok" by each recipient. As usual, we assume that all participants adhere to the pragmatic rules, in particular we assume that they only answer truthfully.

The communicative steps always leave an effect. We concentrate here on the accumulation of knowledge and leave the message scheduling out of consideration. We will however later see that certain protocols are more apt than others for the accumulation of knowledge. The announcement "!φ" as well as the answers to the question "?φ", if received by b via a channel C allow b to eliminate certain epistemic alternatives. Thus, if a formal model is required, it will be a dynamically changing Kripke-frame. However, it is not necessary to spell out the details to make the arguments clear.

In what follows I shall be concerned only with knowledge of nonmodal propositions, as it is not subject to change by rounds of communication. Thus, the formula φ unless otherwise indicated is assumed to be nonmodal.

There are basically two ways in which joint knowledge can become common knowledge. The first is described in Balbiani et al. (2008). Some speaker, a, sends out the message "$!\varphi$" through the channel $H \in \mathscr{C}$. After that, φ is common knowledge for the group H. This is the "push"-method, where someone distributes the knowledge. I should stress that this method is not as straightforward as it appears. In practice, we need to know not only that a sent out "$!\varphi$" via some channel C. It must namely also presupposed that the structure of the network is common knowledge. To see this, think about some newsletter broadcast through the net by some administration. Suppose I get that email and wonder whether a also got it. This in turn requires that I know whether a is part of the email-list address to which this message was sent. (The possibility of registering black carbon copies in email messages complicates the picture a bit. Basically, a recipient of an email knows about all recipients except other black carbon copy recipients.) Additional worries may be whether or not a has actually read and understood the message. Even face to face communication is not innocent in that respect. Even if there is no logical addressing mechanism involved, people can hear the message only if they are close enough, for example. And we may not always know who is within hearing distance (think your house and someone in an adjacent room, or even wiretapping). It is therefore far from clear who physically gets the message; that is, it is not clear what channel is actually being used.

Once all that is granted, however, as is done in this framework, then the broadcast really turns the message into common knowledge among the members of the channel *as long as the return acknowledgement is sent through that same channel as well*. The second method is where some a wants to know whether φ holds and sends out a request, "$?\varphi$", through the channel H. This is the "pull"-method. It turns out, though, that getting an answer to one's question is not that easy. One problem is that the channel might not reach everyone from the intended group H, so that what we get is not what the entire set of agents know, but something weaker. The network structure plays an important role in how we can gain access to knowledge. I shall ignore these complications in the sequel.

To start we make even more drastic simplification and assume that each subset $H \subseteq G$ is a channel. To see that even in this simplified scenario matters are still not so trivial, let us assume that b knows that p_0, but not whether p_1, while c knows that p_1 but not whether p_0, and a wants to know whether or not $p_0 \wedge p_1$ is true. If a simply sends out the request "$?(p_0 \wedge p_1)$" through the channel $\{a, b, c\}$ then he would get no further. Neither b nor c are in a position to answer his request and reply with "don't know". However, if a sends out *two* requests, say "$?p_0$" followed by "$?p_1$", he will reach his goal. b answers "yes" to his first request and c answers "don't know", while b answers "don't know" to the second request, while c answers "yes". After all this is done, a knows that $p_0 \wedge p_1$. Moreover, if the replies are sent through the same channel, b and c also know this. For then c knows that b answered the question "$?p_0$" by "yes", and b knows that c answered the question "$?p_1$" by "no". If furthermore the senders and channels of the messages are common knowledge, then $p_0 \wedge p_1$ becomes common knowledge of $\{a, b, c\}$.

Consider now a second scenario. b knows that $\neg p_0 \vee p_1$, c knows that $p_0 \vee \neg p_1$ and d knows that $p_0 \vee p_1$. In this situation, asking either "$?p_0$" or "$?p_1$" gets a no further. None of the others can answer positively or negatively to these questions. It seems then that what a must ask depends on what the others know. Fortunately, the situation is not that bad. Here is a strategy that always works.

Let "φ" be the formula about which a wants to know whether it is true. Consider a conjunctive normal form δ of φ. This is a conjunction $\delta = \bigwedge_{j\in n}\chi_j$ of maximal disjunctions χ_j. A *maximal disjunction* is a formula of the form st_P, where P is a subset of the set $\mathrm{Var}(\varphi)$ of variables of φ:

(3) $$\mathrm{st}_P := \bigvee_{p\in P} p \vee \bigvee_{p\in \mathrm{Var}(\varphi)-P} \neg p$$

Now suppose that $\varphi \leq \mathrm{st}_P$, that is, that φ implies st_P. Then if I know φ I also know st_P. Moreover, by standard modal principles (distribution of K_a over conjunction),

(4) $$K_a\varphi \leftrightarrow \bigwedge_{j\in n} K_a\chi_j$$

Hence, to obtain knowledge of φ it is enough if I obtain knowledge of every maximal disjunction implied by φ.

Let's consider such a disjunction st_P. If a asks b about st_P, the following may occur: b answers "yes" if b knows that st_P, b answers "no" if b knows that $\neg\,\mathrm{st}_P$, and "don't know" otherwise. What however are circumstances in which b knows neither st_P nor $\neg\,\mathrm{st}_P$ for any P? These are circumstances in which the knowledge of b concerning the variables $\mathrm{Var}(\varphi)$ is zero, that is, if τ is a formula in the variables of $\mathrm{Var}(\varphi)$ that is known by b, then τ is a tautology. For if b does not know $\neg\,\mathrm{st}_P$, then some alternative world does not satisfy $\neg\,\mathrm{st}_P$. That is, some alternative satisfies st_P. If this is the case for all $P \subseteq \mathrm{Var}(\varphi)$, b in effect knows nothing. Thus, as long as b knows something, he can answer "yes" or "no" to some of a's questions.

It follows after some reflection that the following strategy works for a independently of what the other agents know. For all subsets $P \subseteq \mathrm{Var}(\varphi)$ such that $\varphi \leq \mathrm{st}_P$ a needs to send out the question "$?\,\mathrm{st}_P$". If he gets the reply "yes" at least once, st_P is jointly known. If no recipient answers "yes", st_P (and therefore φ) is not jointly known. φ is jointly known if (and only if) every such disjunct is jointly known.

Notice that the answer "no" played a subordinate role. Indeed, b will answer "no" just in case his epistemic alternatives all satisfy $\neg\,\mathrm{st}_P$. In that case, the joint knowledge (since it is not inconsistent) is exactly $\neg\,\mathrm{st}_P$. For a he could reach that conclusion also by looking at the "yes" answers of b: b will answer "yes" to all st_Q where $Q \neq P$. Hence the above communication game can also be played with the following convention. There are only two answers to "$?\varphi$": "yes", when the addressee does know that φ, and "no", when the addressee does not know that φ (but it is unclear whether or not he knows $\neg\varphi$). Even more can be concluded: the strategy works even when a does not know what the answer "no" factually means. The only thing that a needs to know is that "yes" means that the addressee knows that φ. (This situation is not uncommon. It is very often not clear whether people simply deny a claim or whether they wish to assert its falsity.)

4 Network Structure

The structure of the network has been assumed to be trivial, namely the powerset of G. What if that is not the case? Let us go back to the initial scenario where a sends out the request "?φ" through the channel H. This may be interpreted as a request to get to know whether or not φ is joint knowledge *for the group H only*. But mostly a simply intends to get an answer but cannot reach everyone through a channel. Such is the case if $H \notin \mathscr{C}$. There are two ways to look at the matter. The first option is that a is indeed interested in knowing what the group H knows. In that case he can simply send out the request "?$J_H \varphi$", thus indicating that he wishes to know whether or not φ is joint knowledge of the group H. This requires that knowledge operators are transitive, however, since a asks what the individuals know about the joint knowledge of φ not about their knowledge of φ directly. Let us grant however that knowledge is transitive. It is to be seen whether that is a solution to a's predicament. Let us consider the case $\varphi = \mathrm{st}_P$. Suppose b is asked to answer "?$J_H\, \mathrm{st}_P$". If st_P is not an epistemic alternative for b, b knows that $\neg\,\mathrm{st}_P$, and therefore he also knows that $\neg J_H\, \mathrm{st}_P$ if $b \in H$. (If $b \notin H$, he has no first hand knowledge of $J_H\, \mathrm{st}_P$, but may acquire it in the communication process.) So he will answer "yes". In the other case, the answer may be "no" or "don't know", depending on how much b knows about other people's knowledge. a can thus obtain full knowledge about $J_H \varphi$.

This shows how a can find out about what is jointly known by some group. This runs into difficulties, however, as soon as the group H is not a channel or $a \notin H$. Clearly, this can be the case. For example, let $H = \{a_i : i < n\}$ and the network only has the channels $\{a_i, a_{i+1 \bmod n}\}$ (so the network is a cycle of length n) and $a = a_0$. In this case a can only send messages to a_1 and a_{n-1}, but not to, say a_2, if $n > 3$. In this situation, a_0 needs to rely on the willingness of the others to complete the task. To achieve this, we need to change the protocol.

Specifically, we need to assume that when a_0 sends a query "?$J_H \varphi$" to a_1 and a_1 cannot reply "yes", then a_1 will take up the matter and ask around to find out more. So, a_1 will ask in particular a_2 who either knows the answer or goes to ask a_3, and so on. This looks like a valid algorithm. However, it has a drawback. There is no guarantee that it terminates. Initially, one may think that once the request took a full round to finally reach a_0, a_0 could simply interrupt the chain and not send out any more requests. However, some messages might bypass a_0. To see this, let me change the network a little bit. Let $\mathscr{C} := \{\{a_i, a_{i+1 \bmod n}\} : i < n\} \cup \{\{a_{n-1}, a_i\} : i < n-1\}\}$. Suppose the query moves around the circle and finally reaches a_{n-1}. If a_{n-1} does not know the answer, he will contact one of the a_i, and so set the entire chain once again in motion.

Further problems concern the fact that since everyone is allowed to issue a request it is not clear whether the request for "?φ" that reaches a_0 is actually a follow-up to a request he initiated (rather than a_1 or a_2). In the absence of an external scheduling mechanism, calls into the network will not die out if everyone is maximally cooperative. An example is where everyone knows that $p_0 \leftrightarrow p_1$, but does not know whether p_0 (and p_1) or $\neg p_0$ (and therefore $\neg p_1$). If someone issues the request ?p_0, the algorithm will run forever. Still, the surprising fact is that if a is chosing his requests carefully enough, termination is guaranteed. Let the protocol for queries of the form "$?J_H\varphi$" be as follows. If $b \notin H$, b will not give an answer and instead issue the same query to all channels, unless b knows the answer offhand to be "yes" or "no". (Here we take advantage of the communication, because answers to queries force updates across the network.) If $b \in H$ and the answer to the query is "yes" or "no" (because of this epistemic alternatives), that answer is sent and no further action is taken. In the remaining case, b will not send out this answer and instead send out "$?J_H\varphi$" to all channels. Upon receiving the answer "yes" or "no", b will answer back to a with that same answer. This means that the answer "don't know" is in fact never used.

Call H totally connected if for every a and b there is a chain of channels connecting a and b. Alternatively, let $a\ V_{\mathscr{C}}\ b$ if there is a $C \in \mathscr{C}$ such that $a, b \in C$. H is totally connected if and only if $V_{\mathscr{C}}^* = H^2$.

Theorem 3. *Let G be totally connected and $H \subseteq G$. Assume that φ is jointly known by H. The maximally cooperative protocol for "$?J_H\varphi$" terminates if for all $P \subseteq \mathrm{Var}(\varphi)$ sender sends out the request "$?J_H\,\mathrm{st}_P$" for all $\mathrm{st}_P \geq \varphi$ in addition.*

Proof. Here is the catch. Suppose that st_P is true in every epistemic alternative for b. Then b knows that st_P and he will answer the request "$?J_H\,\mathrm{st}_P$" with "yes". His answer will get known to the entire channel to which the request has been sent. $\neg\,\mathrm{st}_P$ will cease to be an alternative for members of that channel. Thus, effectively, after a few rounds $\neg\,\mathrm{st}_P$ will be eliminated throughout H. The protocol will then require termination. If all requests are sent out, and φ is jointly known, then at some point all alternatives incompatible with φ will eventually be eliminated. At this point the answer to the question becomes known to everyone. □

This is reminiscent of the muddy children paradox. The more answers appear the more knowledge is acumulated and allows to give answers to questions to which no helpful answer existed before. The glitch here is that a clever initialisation by a can help to make even the maximally cooperative process terminate without scheduling "from above". However, the problem is that for this algorithm to terminate we need that φ is known. We cannot eliminate it. Suppose for example that $\varphi = p_0$ and no one in the entire network knows either p_0 or $\neg p_0$. Then the algorithm never terminates because no one is in a position to answer the request.

To remedy this, we propose a different algorithm. Instead of asking "$?J_H\,\mathrm{st}_P$", a sends out the requests "$J_{\{b\}}\,\mathrm{st}_P$" for every $b \in H$. Since b can always answer this question, this is garanteed to terminate. The proof is now easy. Since b can be reached (by connectedness) the request will eventually reach b provided that all members of the network try to pass on requests to as many members as they are connected to.

Theorem 4 (Guaranteed Termination)**.** *Let G be totally connected and $H \subseteq G$. The maximally cooperative protocol for "$?J_H\varphi$" terminates if for all $P \subseteq \mathrm{Var}(\varphi)$ and all $b \in H$ sender sends out the request "$?J_{\{b\}}\,\mathrm{st}_P$" in addition.*

Consider again the query "$?J_H p_0$" in a network where no one knows p_0 or $\neg p_0$. In this situation, b will respond "don't know" to the question $?J_b p_0$", and also to the question "$?J_b\neg p_0$". From this one can infer that both p_0 and $\neg p_0$ are possibilities for

b. The algorithm terminates for the simple reason that there is no supposition that anyone other than b himself will know more about what b knows. That is to say, we assume that the protocol will not make b send out a request for help on questions about his own knowledge.

Notice that the “envelope” J_H and $J_{\{b\}}$ is crucial in allowing the participants to route the requests. At the same time—because the message is interpreted as given verbatim—they distort the original query because they ask about what the individuals know to be their knowledge rather than asking about their knowledge directly. In other words, we assume that knowledge satisfies S4.

Finally, there is a different solution to the problem. Change the protocol as follows. On receiving the request “$?J_H\varphi$” an agent b will do the following. If b knows the answer he will reply. Otherwise he will send out the request “$J_{H-\{b\}}\varphi$” to all channels, provided that this is not empty. However, $H - \{b\} = \varnothing$ exactly when $H = \{b\}$. In that case, b will give the answer as best as he can. I call this the *group distribution protocol*.

Theorem 5 (Guaranteed Termination)**.** *Let G be totally connected and $H \subseteq G$. The group distribution protocol for “$?J_H\varphi$” terminates if for all $P \subseteq \mathrm{Var}(\varphi)$ and all $b \in H$ sender sends out the request “$?J_{\{b\}}\,\mathrm{st}_P$” in addition.*

How can we see that this is correct? At first, the query “$J_H\,\mathrm{st}_P$” will be sent out into the network and will distribute itself unchanged until it reaches some $a \in H$. This will then create another query, namely “$?J_{H-\{a\}}\,\mathrm{st}_P$”. And so on, until a query of the form “$?J_{\{b\}}\,\mathrm{st}_P$” is issued that eventually reaches b. b will answer the query, and the answer will propagate through the network until everyone knows it. At this point the query “$?J_{\{b\}}\,\mathrm{st}_P$” will no longer be propagated and will die out. When finally all such queries have been propagated, the answer to “$?J_H\,\mathrm{st}_P$” becomes known throughout the network, and the algorithm terminates. When this has happened for all st_P, the answer will be known for φ as well.

5 Conclusion

This paper is a modest attempt to characterise the notion of joint knowledge and show how agents can find out whether a proposition is or is not jointly known by a group. In closing, I would like to point out some wider significance of this endeavour.

Humans are thirsty for knowledge. Research or daily experience both continue to give us new insights and knowledge. Thus, it is not to be expected that all joint knowledge can one day become common knowledge given enough communication. What is more, there is so much accumulated knowledge that it is not even possible to store all knowledge everywhere. Thus, we seek to distribute the knowledge in a network so as to share the burden of storing it. There is no difference in principle between a bunch of humans and a server farm, in fact. There is a tradeoff between distributing knowledge in a network and storing it at each location separately. Similarly, as humans we need to balance knowing something by heart and having it available from somewhere on need. The terms of the tradeoff are not logical: I have shown how we can get the desired answer. The tradeoff is in terms of effort, of which I have said nothing above. I shall leave that to another occasion.

References

Balbiani, Philippe, Alexandru Baltag, Hans van Ditmarsch, Andreas Herzig, Tomohiro Hosi, and Santiago de Lima. 2008. 'Knowable' as 'Known After an Announcement'. *The Review of Symbolic Logic* 1:305–334.

Brandt, Christoph, and Marcus Kracht. 2011. Syntax, Semantics and Pragmatics in Communication. In *Pragmatic Web Track, 7th International Conference on Semantic Systems, Graz, 7.-9. September 2011*, 195–202.

Fagin, Ronald, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. 1995. *Reasoning about knowledge*. MIT Press.

Goldblatt, Robert I. 1987. *Logics of Time and Computation*. Number 7 in CSLI Lecture Notes. CSLI.

Halpern, Joseph Y. 1987. Using Reasoning About Knowledge to Analyze Distributed Systems. In *Annual reviews of computer science*, ed. J. F. Traub, B. G. Grosz, B. W. Lampson, and N. J. Nilsson, volume 2, 37–68.

Hintikka, Jaakko. 1962. *Knowledge and Belief. An introduction into the logic of the two notions*. Ithaca: Cornell University Press.

Passy, Solomon, and Tinko Tinchev. 1991. An essay in combinatory dynamic logic. *Information and Computation* 93:263–332.

Williamson, Timothy. 2000. *Knowledge and its Limits*. Oxford University Press.