The Relative Expressivity of Public and Private Communication in BMS Logic

Bryan Renne

ABSTRACT. Dynamic Epistemic Logic (DEL) is the study of formal logics that reason about knowledge change. In DEL research, events that cause changes in knowledge are called *updates*. One class of updates—the *BMS updates* due to Baltag, Moss, and Solecki—has received much attention in the DEL literature because of the joint expressive power of the various *BMS Logics* based on these updates. There is, however, very little known about the relative expressive power of the BMS Logics based on individual BMS updates.

The purpose of the present paper is two-fold. First, we provide a succinct, self-contained exposition of the rather complicated syntax and semantics for the class of BMS Logics. Second, we study the relative expressivity of BMS Logics for public and private communication. The results we obtain are a first-step in a larger study whose aim is to characterize the relative expressive power of BMS Logics in general.

1 Introduction

In the epistemic reading of multi-modal logic, we assume that a complete description of a certain moment in time is given by a multi-agent Kripke model [FHMV95]. Kripke models, as we all know, consist of a nonzero number of *worlds*—each corresponding to a propositional model (that is, a truth assignment)—along with various binary relations, one for each agent, that may or may not hold between two given worlds. The binary relations represent an agent's uncertainty: if world Δ is related to world Γ by agent i's relation, then agent i will consider it possible that the actual world is Δ whenever the world is in fact Γ. So for agent i to know something at world Γ, that something must be true at Γ and at all those worlds i considers to be possible at Γ. This is just Hintikka's notion of knowledge [Hin62].

While we have said that a Kripke model is a complete description of a certain moment in time, we have not yet said how time progresses from one moment to the next. In Dynamic Epistemic Logic (DEL), time progresses based on the occurrence of certain events called *updates*. An update is just a function that maps us from one moment to the next moment (that is, from one Kripke model to another Kripke model). To study a particular update π, the approach in DEL is to extend the language by introducing a new modal—let us write it as $[\pi]$ for the moment—whose meaning is to execute the update π. So if φ is a formula, then the new formula $[\pi]\varphi$ has the informal reading "φ holds after the update π occurs." Semantically, the formula $[\pi]\varphi$ is said to be true at world Γ of the Kripke model M if and only if the following holds: if π can be executed at Γ in model M, then φ is true at the world $\pi(\Gamma)$ in the model $\pi(M)$. Here $\pi(M)$ is the Kripke model that results from applying the update π to the model M, and $\pi(\Gamma)$ is the world in $\pi(M)$ that we are taken to when π is executed at the world Γ in M. So we see that $[\pi]\varphi$ expresses a before-after relationship with respect to the update π: if π can be executed, then its execution leads to a new situation in which φ is true.

Work on updates goes back to Plaza [Pla89] and Gerbrandy [Ger99], who independently defined the *public announcement* update, which acts as a form of public communication to all agents in the Kripke model. Baltag, Moss, and Solecki (BMS) extended the Plaza-Gerbrandy work by developing what we call the *BMS updates* [BMS98, BM04, BMS05], which have become quite popular in the DEL literature. BMS updates are structurally complicated and, since each collection of BMS updates yields a logical language describing the updates in that collection, we are led to an even more complicated hierarchy of logical languages. *BMS Logic* is the name we give to the family of all logics that are based on these logical languages. We parameterize these logics by the BMS updates that are described in the language of a given logic.

The first task of this paper is to present a succinct, self-contained overview of the complicated syntax and semantics of BMS Logic. Of notable omission in this overview are the axiomatic systems for the various BMS Logics.[1] The reason for this omission is that our interest later in the paper will be to study issues of language expressivity (which will not involve axiomatics). In particular, we will study the relative expressivity of the BMS Logic for public communication, the BMS Logic for private communication, and the BMS Logic for disguised private communication. We will see that public communication and private communication are expressively incomparable, while disguised private communication is strictly more expressive than both public and private communication. Taken together, our results extend previous expressivity work in [Pla89, Ger99, BM04, BMS05] and may be viewed as a first-step in a larger study whose goal is to provide a general charac-

[1] The interested reader should consult [BM04, BMS05] for an axiomatic study of BMS Logic.

terization of the relative expressivity of the various BMS Logics as a whole. With this in mind, let us proceed by introducing the language of BMS Logic.

2 The Language of BMS Logic

The language of BMS Logic is the extension of the language of n-agent epistemic logic obtained by admitting formula closure under certain modals that we call *BMS modals*.

DEFINITION 1. Given a language $\mathcal{L}$ and a modal $[\pi]$, the language *obtained (from $\mathcal{L}$) by admitting formula closure under* $[\pi]$ is the language $\mathcal{L}'$ whose rules of formula formation are those of $\mathcal{L}$ in addition to the following: if φ is an $\mathcal{L}'$-formula, then $[\pi]\varphi$ is also an $\mathcal{L}'$-formula.

BMS modals are defined relative to certain finite structures we call *BMS frames*.

DEFINITION 2 (Adapted from [BM04]). For a positive integer n, an *n-agent BMS frame*is a tuple $(W, \{S_i\}_{i=1}^n, d)$, where for some integer $m \geq 1$, we have

- W is the (nonempty) set $\{1, 2, 3, \dots, m\}$ of the first m positive integers—we will occasionally refer to the members of W as *worlds*;
- each S_i is a binary relation on W; and
- d is an integer satisfying $0 \leq d \leq m$.

We will omit mention of "n-agent" when n is clear from context. Notice that m is just $|W|$, the size of W.

A BMS frame acts as a schema for formation of BMS modals in the following sense.

DEFINITION 3. Let $\mathcal{L}$ be a language and $B = (W, \{S_i\}_{i=1}^n, d)$ be a BMS frame. If $\{\psi_i\}_{i=1}^d$ is a sequence of $\mathcal{L}$-formulas and $a \in W$, then we call $[\{\psi_i\}_{i=1}^d]^a$ a *BMS modal (based on BMS frame B in language $\mathcal{L}$).* Convention: if $d = 0$, then $\{\psi_i\}_{i=1}^d$ denotes the empty sequence. We use the symbol ϵ for the empty sequence, so a BMS modal based on a BMS frame with $d = 0$ has the form $[\epsilon]^a$.

REMARK 4. Assume the notation of Definition 3 and let $m = |W|$. It is helpful to picture the BMS modal $[\{\psi_i\}_{i=1}^d]^a$ as compact description of a function that maps the integers in W to $\mathcal{L}$-formulas according to the following diagram:

$$\begin{array}{cccccccccccc}
W = \{ & 1 & 2 & 3 & \cdots & d-1 & d & d+1 & d+2 & \cdots & m & \} \\
 & \mapsdown & \mapsdown & \mapsdown & \dots & \mapsdown & \mapsdown & \mapsdown & \mapsdown & \dots & \mapsdown & \\
 & \psi_1 & \psi_2 & \psi_3 & \cdots & \psi_{d-1} & \psi_d & \top & \top & \cdots & \top &
\end{array}$$

Here we have assigned the propositional constant $\top$ (truth) to each integer $x \in W$ that is strictly larger than d (meaning $x > d$). The reason we choose $\top$ instead of some other $\mathcal{L}$-formula will be made clear after we have introduced the semantics. (Notice that if $d = 0$, then our convention in Definition 3 has us assign $\top$ to every integer $x \in W$.) The superscript $a \in W$ in the BMS modal $[\{\psi_i\}_{i=1}^d]^a$ acts to distinguish the $\mathcal{L}$-formula in the a-th coordinate. We will say more on this in the section on the semantics of BMS Logic.

We are interested in extensions of multi-modal epistemic logic obtained by admitting formula closure under the BMS modals that are based on any one of a finite collection of BMS frames. (The restriction to a finite collection is both to keep things relatively simple and also to ensure that our languages are countable.) We thus view the language of BMS Logic as a family of languages parameterized by the fixed collection of BMS frames on which BMS modals are based. We now give a name for this fixed collection of BMS frames.

DEFINITION 5 (Adapted from [BM04]). For a positive integer n, an *n-agent signature* is a finite indexed set $\{B_j\}_{j\in J}$ of n-agent BMS frames.[2] (So J is a finite set.) We will omit mention of "n-agent" when n is clear from context.

Since an n-agent signature $\{B_j\}_{j\in J}$ contains a number of BMS frames, it will be important to indicate the frame B_j on which a given BMS modal is based. To do this, we will add the subscript j to the BMS modals based on B_j. This leads us to the notion of a BMS modal *based on a signature* (as opposed to a BMS modal *based on a BMS frame*, Definition 3).

DEFINITION 6. Let $\mathcal{L}$ be a language and $\mathcal{B} = \{(W_j, \{S_{j,i}\}_{i=1}^n, d_j)\}_{j\in J}$ be an n-agent signature. If $\{\psi_i\}_{i=1}^{d_j}$ is a sequence of $\mathcal{L}$-formulas and $a \in W_j$, then we call $[\{\psi_i\}_{i=1}^{d_j}]_j^a$ a *BMS modal (based on signature $\mathcal{B}$ in language $\mathcal{L}$).* Convention (as in Definition 3): $\{\psi_i\}_{i=1}^{d_j}$ denotes ϵ if $d_j = 0$.

We may now define the language of BMS logic based on a signature.

DEFINITION 7 (Adapted from [BM04]). Let $\mathcal{L}$ be a language and $\mathcal{B}$ be an n-agent signature. $\mathcal{L}(\mathcal{B})$, *the language of BMS Logic (based on signature $\mathcal{B}$ and language $\mathcal{L}$)*, is obtained from $\mathcal{L}$ by admitting formula closure under each BMS modal based on signature $\mathcal{B}$ in language $\mathcal{L}(\mathcal{B})$.[3] (Notice that

[2] Saying that C is a *finite indexed set* means that there is a finite set J and a bijection $f : J \to C$. J is the *index set* or set of *indicies*, and each $j \in J$ is an *index*. f maps each index j to the unique member $f(j)$ of C that is indexed by j, and so $f(j)$ is called the *j-th* member of C.

[3] Thus $\mathcal{L}(\mathcal{B})$ is the language whose rules of formula formation are are those of $\mathcal{L}$ in

$\mathcal{L}(\mathcal{B})$ contains formulas that have BMS modals nested inside other BMS modals.)

NOTATION 8 ($\vec{\chi}$). Since it is a nuisance to use so many symbols in writing the sequence of formulas appearing in a BMS modal, we adopt the following notational conventions.

- $\vec{\chi}$ abbreviates a finite (possibly empty) sequence $\{\chi_i\}_{i=1}^m$.
- A BMS modal based on a signature $\{(W_j, \{S_{j,i}\}_{i=1}^n, d_j)\}_{j\in J}$ will be written as $[\vec{\psi}]_j^a$, which means that $m = d_j$ for the sequence $\{\psi_i\}_{i=1}^m$ abbreviated by $\vec{\psi}$.
- If the sequence $\vec{\psi}$ is of length one, we will write the BMS modal $[\vec{\psi}]_j^a$ as $[\psi]_j^a$, and the symbol ψ will represent the first (and only) formula in the sequence $\vec{\psi}$.

In Definition 7, we defined the language $\mathcal{L}(\mathcal{B})$ of BMS Logic in terms of two parameters: a signature $\mathcal{B}$ and a language $\mathcal{L}$. Before we introduce the semantics for $\mathcal{L}(\mathcal{B})$, we identify three languages $\mathcal{L}$ that will be of particular interest in this paper. (Actually, we are most interested in the second two; we define the first for reasons of concreteness.)

DEFINITION 9. The *language of propositional logic* consists of the propositional constants $\top$ (truth) and $\bot$ (falsity), a countable collection of propositional letters, and symbols for the Boolean connectives (note that we use $\supset$ for implication). The *atoms* consist of the propositional constants and the propositional letters. The *propositional formulas* are built up from the atoms using the Boolean connectives.

DEFINITION 10 ($\mathcal{L}^n$). Let n be a positive integer. $\mathcal{L}^n$, *the language of n-agent epistemic logic*, is obtained from the language of propositional logic by admitting formula closure under each of the modals $K_1, K_2, K_3, \ldots, K_n$.

DEFINITION 11 ($\mathcal{L}_C^n$). Let n be a positive integer. $\mathcal{L}_C^n$, *the language of n-agent epistemic logic with common knowledge*, is obtained from $\mathcal{L}^n$ by admitting formula closure under the modal C.

For readability, we find it useful to introduce the following notation for dual modals.

NOTATION 12. Fix a language $\mathcal{L}_C^n(\mathcal{B})$. The modal $\hat{K}_i$ abbreviates $\neg K_i \neg$, the modal $\hat{C}$ abbreviates $\neg C \neg$, and the modal $\langle\vec{\psi}\rangle_j^a$ abbreviates $\neg[\vec{\psi}]_j^a\neg$.

---

addition to the following rule: if $\{\psi_i\}_{i=1}^{d_j}$ is a sequence of $\mathcal{L}(\mathcal{B})$-formulas, φ is an $\mathcal{L}(\mathcal{B})$-formula, and $a \in W_j$, then $[\{\psi_i\}_{i=1}^{d_j}]_j^a\varphi$ is also an $\mathcal{L}(\mathcal{B})$-formula.

Finally, we define a notion of depth for formulas in the language of BMS Logic. Our notion of formula depth counts the maximum nested depth of modals in a way that ensures that the formula $[\vec{\psi}]^a_j\varphi$ is of strictly greater depth than each of its immediate subformulas.[4]

DEFINITION 13. Let $\mathcal{B}$ be a fixed n-agent signature and φ be an $\mathcal{L}^n_C(\mathcal{B})$-formula. The *depth* of φ, written $d(\varphi)$, is given by induction on the construction of φ as follows.

$$\begin{array}{lcl} d(p) & := & 0, \text{ for } p \text{ an atom} \\ d(\chi \supset \psi) & := & \max(d(\chi), d(\psi)) \\ d(K_i\psi) & := & 1 + d(\psi), \text{ for } 1 \leq i \leq n \\ d(C\psi) & := & 1 + d(\psi) \\ d([\vec{\psi}]^a_j\chi) & := & 1 + d(\chi) + \sum_{i=1}^{d_j} d(\psi_i) \end{array}$$

(Note: we set $\sum_{i=1}^{0} d(\psi_i) := 0$.) Other Boolean connectives are handled as is implication (that is, Boolean connectives take the maximum over the depths of immediate subformulas, adding no additional depth). The depths of $\mathcal{L}^n_C$-formulas, $\mathcal{L}^n(\mathcal{B})$-formulas, and $\mathcal{L}^n$-formulas are defined by dropping the appropriate clauses above.

3 The Semantics of BMS Logic

Both $\mathcal{L}^n(\mathcal{B})$-formulas and $\mathcal{L}^n_C(\mathcal{B})$-formulas are interpreted in n-agent Kripke models.

DEFINITION 14. For a positive integer n, an *n-agent Kripke model* is a tuple $(W, \{R_i\}_{i=1}^n, V)$, where

- W is a nonempty set whose elements are called *worlds*,
- each R_i is a binary relation on W, and
- V is a function mapping each world Γ to a (possibly empty) set $V(\Gamma)$ of propositional letters.

Various relational conditions may be imposed on some or all of the R_i's. We will omit mention of "n-agent" when n is either unimportant or else clear from context.

Formulas in the language of BMS Logic are interpreted at model-world pairs.

[4]Here an *immediate subformula* of φ is a formula ψ that appears in the antecedent of the rule of formula formation that builds φ from other formulas (including ψ).

DEFINITION 15. A *model-world pair* is a pair (M, Γ) consisting of an n-agent Kripke model M and a world Γ in M. To say that a model-world pair (M', Γ') is *in* the model M means that $M' = M$.

We now say what it means for a formula in the language of BMS Logic to be true at a model-world pair.

DEFINITION 16 (Adapted from [BM04]). Let (M, Γ) be a model-world pair in the n-agent Kripke model $(W, \{R_i\}_{i=1}^n, V)$. For a formula $\varphi \in \mathcal{L}_C^n(\mathcal{B})$, we write $M, \Gamma \models \varphi$ to mean that φ is *true* at (M, Γ), and we write $M, \Gamma \not\models \varphi$ to mean that φ is not true at (M, Γ). Truth of a formula at a model-world pair is defined by induction on formula construction as follows.

1. $M, \Gamma \models \top$ and $M, \Gamma \not\models \bot$.

2. $M, \Gamma \models p$ means that $p \in V(\Gamma)$, where p is a propositional letter.

3. Boolean connectives are defined in the mathematical meta-language. Example: $M, \Gamma \models \varphi \supset \psi$ means that $M, \Gamma \models \varphi$ implies $M, \Gamma \models \psi$.

4. $M, \Gamma \models K_i\varphi$ means that $\Gamma R_i \Delta$ implies $M, \Delta \models \varphi$, where $1 \leq i \leq n$.

5. $M, \Gamma \models C\varphi$ means that $\Gamma \left(\bigcup_{i=1}^n R_i\right)^* \Delta$ implies $M, \Delta \models \varphi$, where S^* is the reflexive-transitive closure of the relation S.[5]

6. $M, \Gamma \models [\vec{\psi}]_j^a \chi$ has one of two meanings.

   (a) If $a \leq d_j$, it means that $M, \Gamma \models \psi_a$ implies $M[\vec{\psi}]_j, (\Gamma, a) \models \chi$.

   (b) If $a > d_j$, it means that $M[\vec{\psi}]_j, (\Gamma, a) \models \chi$.

   Here the model $M[\vec{\psi}]_j$, called the model *induced by* $[\vec{\psi}]_j^a$, is given by the following construction, called the *BMS (product) update*.[6] Defining the sets

   $$W_j^{\leq d_j} := \{x \in W_j \mid x \leq d_j\} \text{ and } W_j^{> d_j} := \{x \in W_j \mid x > d_j\} \ ,$$

   $M[\vec{\psi}]_j = (W', \{R_i'\}_{i=1}^n, V')$ is defined as follows.

   - $W' := \{(\Gamma, k) \in W \times W_j^{\leq d_j} : M, \Gamma \models \psi_k\} \cup (W \times W_j^{> d_j})$,

[5] For $S \subseteq W \times W$, let $S^0 := \{(x, x) \mid x \in W\}$ and let $S^{i+1} := \{(x, z) \mid (x, y) \in S^i \wedge (y, z) \in S\}$ for each $i \geq 1$. Then the *reflexive-transitive closure* of S, written S^*, is $\bigcup_{i=0}^\infty S^i$.

[6] The superscript a is dropped from the model $M[\vec{\psi}]_j$ because it does not play a role in the construction. Thus for each $a, b \in W_j$, we have that $[\vec{\psi}]_j^a$ and $[\vec{\psi}]_j^b$ induce the same model $M[\vec{\psi}]_j$.

- $(\Gamma, k)R'_i(\Delta, l)$ means both $\Gamma R_i \Delta$ and $k S_{j,i} l$, and
- $V'(\Gamma, k) := V(\Gamma)$.

Truth for formulas in $\mathcal{L}^n_C$ is obtained by dropping Case 6. Truth for formulas in $\mathcal{L}^n(\mathcal{B})$ is obtained by dropping Case 5. Truth for formulas in $\mathcal{L}^n$ is obtained by dropping Cases 5 and 6. To say that a formula φ is *valid* means that φ is true in every model-world pair; a *validity* is a valid formula.

REMARK 17. Assume the notation in Definition 16. Notice that

$$W \times W_j^{>d_j} = \{(\Gamma, k) \in W \times W_j^{>d_j} : M, \Gamma \models \top\} \ .$$

This is the reason that we chose the formula $\top$ (truth) in Remark 4 as the formula to assign to those integers in $W_j^{>d_j}$.

It is well-known that Kripke models may be used to represent situations of knowledge, belief, and uncertainty [FHMV95]. Functions from n-agent Kripke models to n-agent Kripke models are called *updates* because they may be viewed as a change in situation caused by some event. (Example: while waiting for my plane to board, I hear an announcement that my plane is now boarding; this announcement is an update because it is an event that causes a change in situation with respect to my knowledge.) The construction in Definition 16 defines the way in which a BMS modal $[\vec{\psi}]^a_j$ induces an update called the *BMS (product) update*. Since BMS updates involve a relatively complicated construction, it may be helpful to understand the update induced by a BMS modal $[\vec{\psi}]^a_j$ in a stepwise fashion, as follows.

1. Notational preliminaries.
   (a) Let $m := |W_j|$.
   (b) If $N = (W_N, \{R_{N,i}\}_{i=1}^n, V_N)$ is a Kripke model and χ is a $\mathcal{L}^n_C(\mathcal{B})$-formula, then let χ^N be the set of worlds in N at which χ is true:

   $$\chi^N := \{\Gamma \in W_N : N, \Gamma \models \chi\} \ .$$

   When convenient, we will identify χ^N with the submodel

   $$(W'_N, \{R'_{N,i}\}_{i=1}^n, V'_N)$$

   of N given by $W'_N := \chi^N$, $R'_{N,i} := R_{N,i} \cap (W'_N \times W'_N)$, and $V'_N(\Gamma) = V_N(\Gamma)$ for $\Gamma \in W'_N$.
2. For each integer i satisfying $1 \leq i \leq m$, define the formula χ_i by

$$\chi_i := \begin{cases} \psi_i & \text{if } i \leq d_j, \\ \top & \text{if } i > d_j. \end{cases}$$

This gives us a sequence $\{\chi_i\}_{i=1}^m$ of $\mathcal{L}_C^n(\mathcal{B})$-formulas (as in Remark 4 on Page 3).

3. Produce m disjoint copies of the model M.

$$M \times \{1\} \quad \cdots \quad M \times \{d\} \quad M \times \{d+1\} \quad \cdots \quad M \times \{m\}$$

4. Map the i-th copy of M to the submodel χ_i^M defined by the i-th formula χ_i. Place this submodel in position i.

$$\begin{array}{ccccccc} M \times \{1\} & & M \times \{d\} & M \times \{d+1\} & & M \times \{m\} \\ \downarrow & \cdots & \downarrow & \downarrow & \cdots & \downarrow \\ \psi_1^M \times \{1\} & & \psi_d^M \times \{d\} & \top^M \times \{d+1\} & & \top^M \times \{m\} \end{array}$$

Notice that $\top^M = W$.

5. Write (Γ, k) to represent a world Γ in the position-k submodel χ_k^M. This gives us a set W' of worlds:

$$\begin{aligned} W' &= \bigcup_{i=1}^{m} (\chi_i^M \times \{i\}) \\ &= \bigcup_{i=1}^{d_j} (\psi_i^M \times \{i\}) \cup \bigcup_{i=d_j+1}^{m} (W \times \{i\}) \\ &= \{(\Gamma, k) \in W \times W_j^{\leq d_j} : M, \Gamma \models \psi_k\} \cup (W \times W_j^{>d_j}) \ . \end{aligned}$$

6. The relation R_i' connecting a world Γ in the position-k submodel with a world Δ in the position-l submodel is defined component-wise:

$$(\Gamma, k) R_i' (\Delta, l) \text{ means that } \Gamma R_i \Delta \text{ and } k S_{j,i}, l \ .$$

7. The set of propositional letters that are true at world Γ in the position-k submodel is given by the valuation V from the original model M:

$$V'(\Gamma, k) = V(\Gamma) \ .$$

8. $M[\vec{\psi}]_j$ is the resulting model $(W', \{R_i'\}_{i=1}^n, V')$.

So the meaning of $M, \Gamma \models [\vec{\psi}]_j^a \varphi$ is as follows: if χ_a holds at world Γ in model M, then φ holds at Γ in the position-a submodel χ_a^M when we interconnect χ_a^M with the other submodels as in $M[\vec{\psi}]_j$. Written with more notation: $M, \Gamma \models \chi_a$ implies $M[\vec{\psi}]_j, (\Gamma, a) \models \varphi$.

4 Relative Expressivity

In studying the relative expressivity of two languages $\mathcal{L}$ and $\mathcal{L}'$, we are concerned with the following informal question: can one language say something that the other cannot? This question is in essence a question of semantics (after all, $\mathcal{L}$-formulas and $\mathcal{L}'$-formulas in general need not have the same syntactic form). So once we have found a common semantics for $\mathcal{L}$ and $\mathcal{L}'$, we may then ask whether we can map $\mathcal{L}$-formulas to $\mathcal{L}'$-formulas in a way that preserves truth in the common semantics (meaning the image formula is true in a model of the common semantics exactly when its preimage is true in that same model). This gives us a formal understanding of our informal question above. Let us see how this definition looks for the specific case of BMS Logic.

DEFINITION 18. To say that *the $\mathcal{L}^n_C(\mathcal{B})$-formula φ is expressible by the $\mathcal{L}^n_C(\mathcal{B}')$-formula φ' (or that φ' expresses φ)* means that for every model-world pair (M, Γ), we have $M, \Gamma \models \varphi$ exactly when $M, \Gamma \models \varphi'$.

Definition 18 provides us with a sense in which a formula in one BMS language can be said in another BMS language: φ can be said in $\mathcal{L}^n_C(\mathcal{B}')$ exactly when there is a $\mathcal{L}^n_C(\mathcal{B}')$-formula φ' that expresses φ (in the sense of Definition 18). Our understanding of what it means to say that φ *cannot* be said in $\mathcal{L}^n_C(\mathcal{B}')$ is as follows: φ *distinguishes* two model-world pairs (meaning φ is true in one and not true in the other) and yet these two pairs are *indistinguishable* (meaning not distinguished) by any $\mathcal{L}^n_C(\mathcal{B}')$-formula. This provides a sense of the non-expressivity of φ in $\mathcal{L}^n_C(\mathcal{B}')$ by the following considerations. Model-world pairs may be seen as situations (that is, complete descriptions of the universe in a certain moment of time). If we have that φ expresses something true in situation s_1 and that φ expresses something false in another situation s_2, then for a $\mathcal{L}^n_C(\mathcal{B}')$-formula φ' to say the same thing as does φ, the formula φ' itself ought to be true in s_1 and false in s_2. So if situations s_1 and s_2 are indistinguishable to φ', then φ' cannot be saying the same thing as is φ. And if situations s_1 and s_2 are indistinguishable to every $\mathcal{L}^n_C(\mathcal{B}')$-formula, then no $\mathcal{L}^n_C(\mathcal{B}')$-formula says the same thing as does φ. This leads us to the following definition.

DEFINITION 19. To say that *the $\mathcal{L}^n_C(\mathcal{B})$-formula φ is not expressible in $\mathcal{L}^n_C(\mathcal{B}')$* means that for every non-negative integer r, there are model-world pairs (M_1, Γ_1) and (M_2, Γ_2) such that each of the following holds:

1. for every $\mathcal{L}^n_C(\mathcal{B}')$-formula φ' with $d(\varphi') \leq r$, we have $M_1, \Gamma_1 \models \varphi'$ exactly when $M_2, \Gamma_2 \models \varphi'$; and

2. both $M_1, \Gamma_1 \models \varphi$ and $M_2, \Gamma_2 \not\models \varphi$.

In Definition 19, the world-model pairs that serve as counterexamples to the expressivity of φ by a $\mathcal{L}^n_C(\mathcal{B}')$-formula of depth at most r may in fact depend on r. A stronger notion of non-expressivity would require that a single model-world pair act as a uniform counterexample for every r. While we have used the weaker notion in proving the results that appear in this paper, some results may still hold for the stronger notion—an issue that awaits further investigation.

We conclude this section with the definitions of relative expressivity.

DEFINITION 20. To say that $\mathcal{L}^n_C(\mathcal{B})$ is *more expressive* than $\mathcal{L}^n_C(\mathcal{B}')$ means that every $\mathcal{L}^n_C(\mathcal{B}')$-formula is expressed by some $\mathcal{L}^n_C(\mathcal{B})$-formula. To say that $\mathcal{L}^n_C(\mathcal{B})$ and $\mathcal{L}^n_C(\mathcal{B}')$ are *equally expressive* means that each language is more expressive than the other. To say that $\mathcal{L}^n_C(\mathcal{B})$ is *strictly more expressive* than $\mathcal{L}^n_C(\mathcal{B}')$ means that the former is more expressive than the latter and that the latter is not more expressive than the former. To say that $\mathcal{L}^n_C(\mathcal{B})$ and $\mathcal{L}^n_C(\mathcal{B}')$ are *expressively incomparable* means that neither language is more expressive than the other.

5 Relative Expressivity of Public and Private Communication

Though BMS Logic may be viewed as a fragment of PDL [vBvEK06], BMS Logic is itself of interest due to the natural way in which one can specify various complicated updates that mix public communication with varying degrees of private communication. In this section, we will look at the logics based on three signatures: a signature for public communication, a signature for private communication, and a signature for disguised private communication. We will then compare the relative expressivity of the languages based on each of these signatures.

5.1 Public Announcements

Our first signature induces BMS updates that only communicate public information, in a sense we describe in a moment. These updates, called *public announcements*, were studied by Plaza [Pla89] and Gerbrandy [Ger99] before the introduction of BMS Logic.

DEFINITION 21 (Adapted from [BM04]). Let $\mathcal{P}$ be the n-agent signature containing the single BMS frame

$$(\{1\}, \{(1,1)\}_{i=1}^n, 1) \ .$$

That is, the set of worlds is $\{1\}$; for each integer i satisfying $1 \leq i \leq n$, the relation R_i is $\{(1,1)\}$; and $d = 1$.[7] The language $\mathcal{L}^n(\mathcal{P})$, also written

[7] We probably should have written this BMS frame as $(\{1\}, \{\{(1,1)\}\}_{i=1}^n, 1)$. (Note the

PUB^n, is called *the language of public announcement logic (without common knowledge).* The language $\mathcal{L}^n_C(\mathcal{P})$, also written PUB^n_C, is called *the language of public announcement logic with common knowledge.*

A *public announcement* of a formula φ is an update that takes a model M to the submodel φ^M of M defined by φ.[8] In PUB^n (or PUB^n_C), a public-announcement formula has the form $[\varphi]^1_1\psi$ and is given the informal reading "ψ holds after φ is publicly announced." A public announcement is viewed as a public communication by way of analogy: if p is a propositional letter, then the PUB^n_C-formula $[p]^1_1 Cp$ ("p is common knowledge after p is publicly announced") is valid.[9]

5.2 Private Announcements

Just as there is a BMS update for public communication to each of the n agents in an n-agent Kripke model, there is a BMS update for a private communication to exactly one of the n agents.

DEFINITION 22 (Adapted from [BM04]). Given positive integers n and j such that $j \leq n$, the *private announcement to* j, written Pri^n_j, is the BMS frame $(\{1,2\}, \{R_i\}^n_{i=1}, 1)$, where

$$R_i := \begin{cases} \{(1,1),(2,2)\} & \text{if } i = j, \\ \{(1,2),(2,2)\} & \text{if } i \neq j. \end{cases}$$

We may write Pri^n_j without the superscript n when n is clear from context.

The language of *private announcement logic* allows for a private announcement to each agent.

DEFINITION 23. Let $\mathcal{P}r$ be the n-agent signature $\{\mathrm{Pri}^n_j \mid 1 \leq j \leq n\}$, where the j-th BMS frame in this signature is Pri^n_j. The language $\mathcal{L}^n(\mathcal{P}r)$, also written PRI^n, is called *the language of private announcement logic (without common knowledge).* The language $\mathcal{L}^n_C(\mathcal{P}r)$, also written PRI^n_C, is called *the language of private announcement logic with common knowledge.*

Private announcements provide for private communication by way of the following analogy. If p is a propositional letter, then the following PRI^n_C-

extra pair of curly brackets.) However, we find the notation less clunky in this situation when we identify the singleton set $\{(1,1)\}$ with its only element $(1,1)$.

[8] See Item 1b on Page 8 for the definition of φ^M.

[9] Not all formulas become common knowledge after they are announced; in fact, some true formulas become false after they are announced (example: $p \wedge \neg K_1 p$). See [vDK06] for a summary of work done on characterizing the *successful* formulas (those formulas that remain true after they are announced).

formula is valid:

$$\left(\bigwedge_{i=1}^{n} \neg K_i p\right) \supset [p]_j^1 \left(K_j p \wedge \bigwedge_{i \neq j} \neg K_i p\right) \quad .$$

This formula may be read, "if no one knows p, then, after p is privately announced to j, only j knows p."[10]

5.3 Disguised Private Announcements

Finally, let us introduce announcements that can communicate private information to exactly one agent while appearing to the other agents like a public announcement of something else.

DEFINITION 24. Let $\mathrm{Pri}_j^n = (\{1,2\}, \{R_i\}_{i=1}^n, 1)$ be the private announcement to j. Then the *disguised private announcement to* j, written Dis_j^n, is the BMS frame $(\{1,2\}, \{R_i\}_{i=1}^n, 2)$. We may write Dis_j^n without the superscript n when n is clear from context.

Note that the difference between Pri_j^n and Dis_j^n is the last coordinate, which is a 2 in Dis_j^n as opposed to a 1 in Pri_j^n. This difference is crucial, as it allows us to specify a formula other than $\top$ that will define the position-2 submodel in the induced model. (Recall our discussion beginning on Page 8 of the stepwise construction of the induced model.) We will see that this additional formula is the "disguised" public announcement formula.

DEFINITION 25. Let $\mathcal{S}$ be the n-agent signature $\{\mathrm{Dis}_j^n \mid 1 \leq j \leq n\}$, where the j-th BMS frame in this signature is Dis_j^n. The language $\mathcal{L}^n(\mathcal{S})$, also written DIS^n, is called the *language of disguised private announcement logic (without common knowledge).* The language $\mathcal{L}_C^n(\mathcal{S})$, also written DIS_C^n, is called *the language of disguised private announcement logic with common knowledge.*

Disguised private announcements gain their name by way of the following analogy. If p and q are propositional letters, then the following DIS_C^n-formula is valid:

$$[p, q]_j^1 \left(K_j p \wedge \bigwedge_{i \neq j} K_i q\right) \quad .$$

This formula may be read, "after p disguised as q is privately announced to j, we have that j knows p while everyone else has the 'false knowledge' (that is, the belief) that q."[11]

[10] To the author's knowledge, there has not yet been a study of the formulas *successful* for private announcements (the formulas that remain true after they are privately announced). Of course, the same issue can be studied for an arbitrary (BMS) update.

[11] Notice that the second formula ψ in the modal $[\varphi, \psi]_j^a$ only acts as a "disguise" when

5.4 Results on Relative Expressivity

We now turn to our results concerning the relative expressivity of the three languages PUB^n_C, PRI^n_C, and DIS^n_C. We assume $n \geq 2$ in order to avoid a technical pitfall.[12] First a result from [BMS05].

THEOREM 26 (From [BMS05]). *For $n \geq 2$, PUB^n_C is not more expressive than PRI^n_C.*

Proof. It is shown in [BMS05] that the PRI^n_C-formula $\langle p\rangle^1_1 \hat{C}\hat{K}_2\neg p$ is not expressible in PUB^n_C, where p is a propositional letter. ■

Theorem 26 is the technical sense in which we can say something with private announcements that cannot be said with public announcements. Intuitively, this is obvious: private announcements allow us to communicate privately, which is not possible with public announcements. However, this result also works the other way around.

THEOREM 27. *For $n \geq 2$, PRI^n_C is not more expressive than PUB^n_C.*

Proof. We outline the proof that the PUB^n_C-formula $\langle p\rangle^1_1 \hat{C}q$ is not expressible in PRI^n_C, where p and q are propositional letters. A full proof can be found in [Ren07].

We will use a model constructed in [BMS05] for our own purposes.[13] For a non-negative integer r, the n-agent Kripke model C^r is a cycle of size $4r+4$ arranged in a clockwise fashion, so each point has both an outgoing arrow leading to its clockwise-next neighbor and also an incoming arrow coming from its clockwise-previous neighbor. We designate two diametrically opposite points: the top t and the bottom b. The arrows are labeled either 1 or 2 in an alternating pattern, with the arrow outgoing from b a 2-arrow. p is true at all points except for t, and q is true only at b.

We will write (x, y) to denote a pair of diametrically opposite nodes such that $x \neq b$, $x \neq t$, and there is a path σ in C^r from x to t that does not contain b. We write $d(x, y)$ for the length of σ.

Notice that for the PUB^n_C-formula $\langle p\rangle^1_1 \hat{C}q$, we have $C^r, x \not\models \langle p\rangle^1_1 \hat{C}q$ and $C^r, y \models \langle p\rangle^1_1 \hat{C}q$ for each (x, y). The reason: $C^r, t \not\models p$ implies that the node t is omitted in the construction of $C^r[\psi]_1$.

By induction on $d(x, y)$, with $1 \leq d(x, y) \leq 2r+1$, we argue by a sub-induction on the construction of a PRI^n_C-formula φ satisfying $d(\varphi) < d(x, y)$

$a = 1$. If $a = 2$, then $[\varphi, \psi]^a_j$ induces the public announcement of ψ, something we prove later in Theorem 30.

[12] In particular, if $n = 1$ and we restrict ourselves to Kripke models that are reflexive and transitive, then Theorem 26 fails. We expect the same for Theorem 27, though we have not verified this result.

[13] [BMS05] used the model to show PUB^n_C is strictly more expressive than $\mathcal{L}^n_C$ for $n \geq 2$.

that $C^r, x \models \varphi$ iff $C^r, y \models \varphi$. The only interesting part of this induction is the inductive step of the sub-induction that considers a PRI^n_C-formula of the form $[\psi]^a_j \chi$ ("χ is true after a private announcement to j that ψ"). Now we have that $C^r, x \models \psi$ iff $C^r, y \models \psi$ and so we may assume that each side of this iff-statement is true. For a node n in a model M, let $\mathcal{T}(M, n)$ be the tree model generated from the point n in M. It can then be argued that one of two cases obtains. Case one: $\mathcal{T}(C^r[\psi]_j, (x, a))$ is isomorphic to $\mathcal{T}(C^r, x)$ and $\mathcal{T}(C^r[\psi]_j, (y, a))$ is isomorphic to $\mathcal{T}(C^r, y)$, from which it follows by the induction hypothesis in the sub-induction that $C^r[\psi]_j, (x, a) \models \chi$ iff $C^r[\psi]_j, (y, a) \models \chi$. Case two: $\mathcal{T}(C^r[\psi]_j, (x, a))$ and $\mathcal{T}(C^r[\psi]_j, (y, a))$ are each isomorphic to a one-world model in which only p is true, from which the same iff-statement follows immediately.

So by choosing $r = d(\varphi)$ and a pair (x, y) with $d(x, y) > d(\varphi)$, we have that the PRI^n_C-formula φ does not express the PUB^n_C-formula $\langle p \rangle^1_1 \hat{C} q$. ∎

We also ought to expect a result like Theorem 27; after all, the power of public announcements comes from the fact that we can create common knowledge, whereas no finite number of private announcements can achieve common knowledge [PK92].

Theorems 26 and 27 provide us with two important corollaries. Our first, which follows from Theorems 26 and 27, provides a formal sense in which public and private communication are fundamentally different when we have common knowledge.

COROLLARY 28. *For $n \geq 2$, PUB^n_C and PRI^n_C are expressively incomparable.*

Our second corollary provides a formal sense in which private communication (with common knowledge) allows us to say things that cannot be said with (common) knowledge statements alone.

COROLLARY 29. *For $n \geq 2$, PRI^n_C is strictly more expressive than both $\mathcal{L}^n$ and $\mathcal{L}^n_C$.*

Proof. In [BMS05], it is shown that PUB^n_C is strictly more expressive than $\mathcal{L}^n_C$. Combining this result with Theorem 27 and the fact that PRI^n_C is an extension of $\mathcal{L}^n_C$, the result follows. The result for $\mathcal{L}^n$ then follows from the fact that $\mathcal{L}^n_C$ is strictly more expressive than $\mathcal{L}^n$. ∎

Finally, while public and private communication are essentially different, disguised private communication captures both notions at once.

THEOREM 30. *Every PUB^n_C-formula is expressible by some DIS^n_C-formula. Likewise, every PRI^n_C-formula is expressible by some DIS^n_C-formula.*

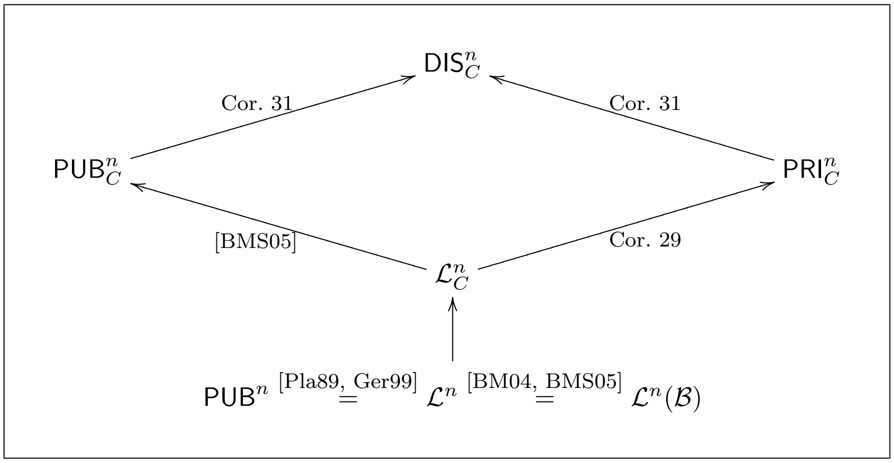

Figure 1. Relative expressivity of various BMS Logics for $n \geq 2$.

Proof. Define a translation $t : \mathsf{PUB}^n_C \to \mathsf{DIS}^n_C$ by $([\psi]^1_1\chi)^t := [\top, \psi^t]^2_1\chi^t$. Define another translation $u : \mathsf{PRI}^n_C \to \mathsf{DIS}^n_C$ by $([\psi]^a_j\chi)^u := [\psi^u, \top]^a_j\chi^u$. See [Ren07] for details. ∎

Theorems 26, 27, and 30 then imply the following corollary.

COROLLARY 31. *For $n \geq 2$, DIS^n_C is strictly more expressive than each of PUB^n_C, PRI^n_C, and $\mathcal{L}^n_C$.*

Figure 1 summarizes our results in context.

6 Conclusions and Directions for Further Study

We have shown that public and private communication are expressively incomparable. This provides a formal sense in which public and private communication are essentially different, something in-line with our intuitions about these communication types. We have also shown that disguised private communication is strictly more expressive than both public and private communication.

Nonetheless, using the phrase *minimal combination of public and private communication* to refer to a smallest theory T such that every T-theorem expresses a PUB^n_C-validity or a PRI^n_C-validity, Corollary 31 suggests that the DIS^n_C-validities are not the minimal combination of private and public communication. Finding this T—which may be just a trivial restriction of the BMS Logic of both public and private communications—would allow us to identify the collection of T-theorems that express both a PUB^n_C-validity and a PRI^n_C-validity, providing a sense in which the PUB^n_C-validities and the

PRI^n_C-validities overlap. Studying this overlap may help us gain a deeper understanding of the relationship between public and private communication: if the overlap expresses just the $\mathcal{L}^n_C$-validities, then we have a sense in which public and private communication are completely different; otherwise, if the overlap expresses more than just the $\mathcal{L}^n_C$-validities, then that part of the overlap that expresses common validities outside the $\mathcal{L}^n_C$-validities is a description of the ways in which public and private communication are the same.

In the broadest sense, this paper is the beginning of a larger study whose aim is to characterize in general terms the relative expressivity of the language $\mathcal{L}^n_C(\mathcal{B})$ and the language $\mathcal{L}^n_C(\mathcal{B}')$. It is the author's hope that there is some natural criterion that holds between the signatures $\mathcal{B}$ and $\mathcal{B}'$ exactly when we have a particular relative expressivity result between the languages based on these signatures. This would solve the relative expressivity questions for BMS Logic all at once and would open the door for more considerations like those of the previous paragraph: what is the minimal combination of the $\mathcal{L}^n_C(\mathcal{B})$-validities and the $\mathcal{L}^n_C(\mathcal{B}')$-validities, and how do these validities overlap? Such questions, like those of the previous paragraph, await further investigation.

BIBLIOGRAPHY

[BM04] Alexandru Baltag and Lawrence S. Moss. Logics for epistemic programs. *Synthese*, 139(2):165–224, 2004.

[BMS98] Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki. The logic of common knowledge, public announcements, and private suspicions. In Itzhak Gilboa, editor, *Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge (TARK VII)*, pages 43–56, Evanston, IL, USA, 1998.

[BMS05] Alexandru Baltag, Lawrence S. Moss, and Sławomir Solecki. Logics for epistemic actions: completeness, decidability, expressivity. Manuscript, 2005.

[FHMV95] Ronald Fagin, Joseph Y. Halpern, Yoram Moses, and Moshe Y. Vardi. *Reasoning about Knowledge*. The MIT Press, 1995.

[Ger99] Jelle Gerbrandy. *Bisimulations on Planet Kripke*. PhD thesis, University of Amsterdam, 1999.

[Hin62] Jaakko Hintikka. *Knowledge and Belief*. Cornell University Press, 1962.

[PK92] Rohit Parikh and Paul Krasucki. Levels of knowledge in distributed systems. In *Sādhanā*, pages 167–191. Indian Academy of Sciences, March 1992.

[Pla89] Jan A. Plaza. Logics of public communications. In Zbigniew W. Ras, editor, *Proceedings of the Fourth International Symposium on Methodologies for Intelligent Systems (ISMIS 1989)*. North-Holland, 1989.

[Ren07] Bryan Renne. The relative expressivity of public and private communication in BMS logic. Technical Report TR-2007012, CUNY Ph.D. Program in Computer Science, 2007.

[vBvEK06] Johan van Benthem, Jan van Eijck, and Barteld Kooi. Logics of communication and change. *Information and Computation*, 204(11):1620–1662, 2006.

[vDK06] Hans van Ditmarsch and Barteld Kooi. The secret of my success. *Synthese*, 151:201–232, 2006.