A dynamic epistemic logic analysis of the equality negation task

Éric Goubault[1], Marijana Lazić[2], Jérémy Ledent[1], and Sergio Rajsbaum[3]

[1] LIX, CNRS, École Polytechnique, Institute Polytechnique de Paris
[2] TU München
[3] Instituto de Matemáticas, UNAM

Abstract. In this paper we study the solvability of the *equality negation* task in a simple wait-free model where processes communicate by reading and writing shared variables or exchanging messages. In this task, two processes start with a private input value in the set $\{0, 1, 2\}$, and after communicating, each one must decide a binary output value, so that the outputs of the processes are the same if and only if the input values of the processes are different. This task is already known to be unsolvable; our goal here is to prove this result using the dynamic epistemic logic (DEL) approach introduced by Goubault, Ledent and Rajsbaum in GandALF 2018. We show that in fact, there is no epistemic logic formula that explains why the task is unsolvable. We fix this issue by extending the language of our DEL framework, which allows us to construct such a formula, and discuss its utility.

Keywords: Dynamic Epistemic Logic · Distributed computing · Equality negation.

1 Introduction

Background. Computable functions are the basic objects of study in computability theory. A function is computable if there exists a Turing machine which, given an input of the function domain, returns the corresponding output. If instead of one Turing machine, we have many, and each one gets only one part of the input, and should compute one part of the output, we are in the setting of *distributed computability*, e.g. [1, 20]. The sequential machines are called *processes*, and are allowed to be infinite state machines, to concentrate on the interaction aspects of computability, disregarding sequential computability issues. The notion corresponding to a function is a *task*, roughly, the domain is a set of input vectors, the range is a set of output vectors, and the task specification Δ is an input/output relation between them. An input vector I specifies in its i-th entry the (private) input to the i-th process, and an output vector $O \in \Delta(I)$ states that it is valid for each process i to produce as output the i-th entry of O, whenever the input vector is I. An important example of a task is *consensus*, where each process is given an input from a set of possible input values, and the participating processes have to agree on one of their inputs.

A *distributed computing model* has to specify various details related to how the processes communicate with each other and what type of failures may occur. It turns out that different models may have different power, i.e., solve different sets of tasks. In this paper we consider the *layered message-passing model* [12], both because of its relevance to real systems, and because it is the basis to study task computability. This simple, wait-free round-based model where messages can be lost, is described in Section 2.

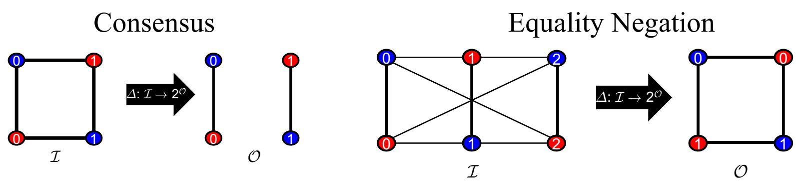
The theory of distributed computability has been well-developed since the early 1990's [15], with origins even before [4, 7], and overviewed in a book [12]. It was discovered that the reason for why a task may or may not be computable is of a topological nature. The input and output sets of vectors are best described as *simplicial complexes*, and a task can be specified by a relation Δ from the input complex $\mathcal{I}$ to the output complex $\mathcal{O}$. The main result is that a task is solvable in the layered message-passing model if and only if there is a certain subdivision of the input complex $\mathcal{I}$ and a certain simplicial map δ to the output complex $\mathcal{O}$, that respects the specification Δ. This is why the layered message-passing model is fundamental; models that can solve more tasks than the layered message-passing model preserve the topology of the input complex less precisely (they introduce "holes").

Motivation. We are interested in understanding distributed computability from the epistemic point of view. What is the knowledge that the processes should gain, to be able to solve a task? This question began to be addressed in [11], using dynamic epistemic logic (DEL). Here is a brief overview of the approach taken in [11]. A new *simplicial complex model* for a multi-agent system was introduced, instead of the usual Kripke epistemic $S5$ model based on graphs. Then, the initial knowledge of the processes is represented by a simplicial model, denoted as $\mathcal{I}$, based on the input complex of the task to be solved. The distributed computing model is represented by an action model $\mathcal{A}$, and the knowledge at the end of the executions of a protocol is represented by the product update $\mathcal{I}[\mathcal{A}]$, another simplicial model. Remarkably, the task specification is also represented by an action model $\mathcal{T}$, and the product update gives a simplicial complex model $\mathcal{I}[\mathcal{T}]$ representing the knowledge that should be acquired, by a protocol solving the task. The task $\mathcal{T}$ is *solvable* in $\mathcal{A}$ whenever there exists a morphism $\delta : \mathcal{I}[\mathcal{A}] \to \mathcal{I}[\mathcal{T}]$ such that the diagram of simplicial complexes below commutes.

$$\begin{tikzcd}
 & \mathcal{I}[\mathcal{A}] \arrow[dl, "\pi"'] \arrow[d, dashed, "\delta"] \\
\mathcal{I} & \mathcal{I}[\mathcal{T}] \arrow[l, "\pi"]
\end{tikzcd}$$

Thus, to prove that a task is unsolvable, one needs to show that no such δ exists. But one would want to produce a specific formula, that concretely represents knowledge that exists in $\mathcal{I}[\mathcal{T}]$, but has not been acquired after running the protocol, namely in $\mathcal{I}[\mathcal{A}]$. Indeed, it was shown in [11] that two of the main impossibilities in distributed computability, consensus [7, 19] and approximate agreement [12], can be expressed by such a formula. However, for other unsolvable tasks (e.g. set agreement), no such formula has been found, despite the fact that no morphism δ exists.

Contributions. In this paper we show that actually, there are unsolvable tasks, for which no such formula exists, namely, the *equality negation* task, defined by Lo and Hadzilacos [18]. This task was introduced as the central idea to prove that the consensus hierarchy [13, 16] is not robust.



Consider two processes P_0 and P_1, each of which has a private input value, drawn from the set of possible input values $I = \{0, 1, 2\}$. After communicating, each process must irrevocably decide a binary output value, either 0 or 1, so that the outputs of the processes are the same if and only if the input values of the processes are different.

It is interesting to study the solvability of the equality negation task from the epistemic point of view. It is well known that there is no wait-free consensus algorithm in our model [5, 19]. The same is true for equality negation, as shown in [18, 10]. This is intriguing because there is a formula that shows the impossibility of consensus (essentially reaching common knowledge on input values) [11], while, as we show here, there is no such formula for equality negation. In more detail, it is well known that consensus is intimately related to connectivity, and hence to common knowledge, while its specification requires deciding unto disconnected components of the output complex. The equality negation task is unsolvable for a different reason, since its output complex is connected. Moreover, equality negation is strictly weaker than consensus: consensus can implement equality negation, but not viceversa (the latter is actually a difficult proof in [18]). So it is interesting to understand the difference between the knowledge required to solve each of these tasks.

Our second contribution is to propose an extended version of our DEL framework, for which there is such a formula. Intuitively, the reason why we cannot find a formula witnessing the unsolvability of the task is because our logical language is too weak to express the knowledge required to solve the task. So, our solution is to enrich the language by adding new atomic propositions, allowing us to express the required formula.

Organization. Section 2 recalls the DEL framework introduced in [11], and defines the layered message-passing model in this context. In Section 3 we study the equality negation task using DEL. First we explain why the impossibility proof does not work in the standard setting, then we propose an extension allowing us to make the proof go through. The long version of this paper [9] includes all proofs, as well as a detailed treatment of the equality negation task following the combinatorial topology approach, for completeness, but also for comparison with the DEL approach.

2 Preliminaries

2.1 Topological models for Dynamic Epistemic Logic (DEL)

We recap here the new kind of model for epistemic logic based on chromatic simplicial complexes, introduced in [11]. The geometric nature of simplicial complexes allows us to consider higher-dimensional topological properties of our models, and investigate their meaning in terms of knowledge. The idea of using simplicial complexes comes from distributed computability [12, 17]. After describing simplicial models, we explain how to use them in DEL.

Syntax. Let At be a countable set of atomic propositions and Ag a finite set of agents. The language $\mathcal{L}_K$ is generated by the following BNF grammar:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \qquad p \in \mathrm{At},\ a \in \mathrm{Ag}$$

In the following, we work with $n+1$ agents, and write $\mathrm{Ag} = \{a_0, \ldots, a_n\}$.

Semantics. The usual semantics for multi-agent epistemic logic is based on Kripke frames. The notion of model that we use here, which is based on simplicial complexes, is merely a reformulation of the usual Kripke models using a different formalism. The benefits of this reformulation is that it makes explicit the topological information of Kripke frames. The precise relationship between the usual Kripke models and our simplicial models is studied thoroughly in [11].

Definition 1 (Simplicial complex [17]). *A* simplicial complex $\langle V, M\rangle$ *is given by a set V of* vertices *and a family M of non-empty finite subsets of V called* simplices*, such that for all $X \in M$, $Y \subseteq X$ implies $Y \in M$. We say that Y is a* face *of X.*

Usually, the set of vertices is implicit and we simply refer to a simplicial complex as M. We write $\mathcal{V}(M)$ for the set of vertices of M. A vertex $v \in \mathcal{V}(M)$ is identified with the singleton $\{v\} \in M$. Elements of M are called *simplices*, and those which are maximal w.r.t. inclusion are *facets* (or *worlds*), the set of which is denoted by $\mathcal{F}(M)$. The *dimension* of a simplex $X \in M$ is $|X| - 1$. A simplicial complex M is *pure* if all its facets are of the same dimension n. In this case, we say M is of dimension n. Given a finite set Ag of agents (that we will represent as colors), a *chromatic simplicial complex* $\langle M, \chi\rangle$ consists of a simplicial complex M and a coloring map $\chi : \mathcal{V}(M) \to \mathrm{Ag}$, such that for all $X \in M$, all the vertices of X have distinct colors.

Definition 2 (Simplicial map). *Let C and D be two simplicial complexes. A* simplicial map $f : C \to D$ *maps the vertices of C to vertices of D, such that if X is a simplex of C, $f(X)$ is a simplex of D. A* chromatic simplicial map *between two chromatic simplicial complexes is a simplicial map that preserves colors.*

For technical reasons, we restrict to models where all the atomic propositions are saying something about some local value held by one particular agent. All the examples that we are interested in will fit in that framework. Let *Val* be some countable set of values, and $\mathrm{At} = \{p_{a,x} \mid a \in \mathrm{Ag}, x \in \mathit{Val}\}$ be the set of *atomic propositions*. Intuitively, $p_{a,x}$ is true if agent a holds the value x. We write At_a for the atomic propositions concerning agent a.

A *simplicial model* $\mathcal{M} = \langle C, \chi, \ell \rangle$ consists of a pure chromatic simplicial complex $\langle C, \chi \rangle$ of dimension n, and a labeling $\ell : \mathcal{V}(C) \to \mathscr{P}(\mathrm{At})$ that associates with each vertex $v \in \mathcal{V}(C)$ a set of atomic propositions concerning agent $\chi(v)$, i.e., such that $\ell(v) \subseteq \mathrm{At}_{\chi(v)}$. Given a facet $X = \{v_0, \ldots, v_n\} \in C$, we write $\ell(X) = \bigcup_{i=0}^{n} \ell(v_i)$. A *morphism* of simplicial models $f : \mathcal{M} \to \mathcal{M}'$ is a chromatic simplicial map that preserves the labeling: $\ell'(f(v)) = \ell(v)$ (and χ).

Definition 3. *We define the truth of a formula φ in some epistemic state $(\mathcal{M}, X)$ with $\mathcal{M} = \langle C, \chi, \ell \rangle$ a simplicial model, $X \in \mathcal{F}(C)$ a facet of C and $\varphi \in \mathcal{L}_K(\mathrm{Ag}, \mathrm{At})$. The satisfaction relation, determining when a formula is true in an epistemic state, is defined as:*

$\mathcal{M}, X \models p$	*if* $p \in \ell(X)$
$\mathcal{M}, X \models \neg\varphi$	*if* $\mathcal{M}, X \not\models \varphi$
$\mathcal{M}, X \models \varphi \wedge \psi$	*if* $\mathcal{M}, X \models \varphi$ *and* $\mathcal{M}, X \models \psi$
$\mathcal{M}, X \models K_a\varphi$	*if for all* $Y \in \mathcal{F}(C), a \in \chi(X \cap Y)$ *implies* $\mathcal{M}, Y \models \varphi$

It is not hard to see that this definition of truth agrees with the usual one on Kripke models (see [11]).

DEL and its topological semantics. DEL is the study of modal logics of model change [3, 6]. A modal logic studied in DEL is obtained by using action models [2], which are relational structures that can be used to describe a variety of communication actions.

Syntax. We extend the syntax of epistemic logic with one more construction:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \wedge \varphi) \mid K_a\varphi \mid [\alpha]\varphi \qquad p \in \mathrm{At},\ a \in \mathrm{Ag}$$

Intuitively, $[\alpha]\varphi$ means that φ is true after some *action* α has occurred. An action can be thought of as an announcement made by the environment, which is not necessarily public, in the sense that not all agents receive these announcements. The semantics of this new operator should be understood as follows:

$$\mathcal{M}, X \models [\alpha]\varphi \quad \text{if} \quad \mathcal{M}[\alpha], X[\alpha] \models \varphi$$

i.e., the formula $[\alpha]\varphi$ is true in some world X of $\mathcal{M}$ whenever φ is true in some new model $\mathcal{M}[\alpha]$, where the knowledge of each agent has been modified according to the action α. To define formally what an action is, we first need to introduce the notion of *action model*. An action model describes all the possible actions that might happen, as well as how they affect the different agents.

A simplicial complex version of DEL. An *action model* is a structure $\mathcal{A} = \langle T, \sim, \mathsf{pre}\rangle$, where T is a domain of *actions*, such that for each $a \in \mathrm{Ag}$, $\sim_a$ is an equivalence relation on T, and $\mathsf{pre} : T \to \mathcal{L}_{\mathcal{K}}$ is a function that assigns a *precondition* formula $\mathsf{pre}(t)$ to each $t \in T$. An action model is *proper* if for any two different actions $t, t' \in T$, there is an agent $a \in \mathrm{Ag}$ who can distinguish between them, i.e., $t \not\sim_a t'$.

Given a simplicial model $\mathcal{M} = \langle C, \chi, \ell\rangle$ and an action model $\mathcal{A} = \langle T, \sim, \mathsf{pre}\rangle$, we define the *product update simplicial model* $\mathcal{M}[\mathcal{A}] = \langle C[\mathcal{A}], \chi[\mathcal{A}], \ell[\mathcal{A}]\rangle$ as follows. Intuitively, the facets of $C[\mathcal{A}]$ should correspond to pairs (X, t) where $X \in C$ is a world of $\mathcal{M}$ and $t \in T$ is an action of $\mathcal{A}$, such that $\mathcal{M}, X \models \mathsf{pre}(t)$. Moreover, two such facets (X, t) and (Y, t') should be glued along their a-colored vertex whenever $a \in \chi(X \cap Y)$ and $t \sim_a t'$. Formally, the vertices of $C[\mathcal{A}]$ are pairs (v, E) where $v \in \mathcal{V}(C)$ is a vertex of C; E is an equivalence class of $\sim_{\chi(v)}$; and v belongs to some facet $X \in C$ such that there exists $t \in E$ such that $\mathcal{M}, X \models \mathsf{pre}(t)$. Such a vertex keeps the color and labeling of its first component: $\chi[\mathcal{A}](v, E) = \chi(v)$ and $\ell[\mathcal{A}](v, E) = \ell(v)$.

Given a product update simplicial model $\mathcal{M}[\mathcal{A}] = \langle C[\mathcal{A}], \chi[\mathcal{A}], \ell[\mathcal{A}]\rangle$ as above, one can naturally enrich it by extending the set of atomic propositions in order to capture the equivalence class of $\sim_{\chi(v)}$ on each vertex v. The extended set of atomic propositions would then be $\widehat{\mathrm{At}} = \mathrm{At} \cup \{p_E \mid E \in T/{\sim_a},\ a \in \mathrm{Ag}\}$, where $T/{\sim_a}$ denotes the set of all equivalence classes of $\sim_a$. In that case, the *extended product update model* is $\widehat{\mathcal{M}[\mathcal{A}]} = \langle C[\mathcal{A}], \chi[\mathcal{A}], \widehat{\ell}[\mathcal{A}]\rangle$, that differs from $\mathcal{M}$ only in labeling. Namely, the enriched labeling $\widehat{\ell}[\mathcal{A}]$ maps each vertex $(v, E) \in C[\mathcal{A}]$ into the set of atomic propositions $\widehat{\ell}[\mathcal{A}]((v, E)) = \ell(v) \cup \{p_E\}$. On this extended model $\widehat{\mathcal{M}[\mathcal{A}]}$, we can interpret formulas saying something not only about the atomic propositions of $\mathcal{M}$, but also about the actions that may have occurred.

In the next section, we describe a particular action model of interest, the one corresponding to the layered message-passing model described in Section 2.2.

2.2 The layered message-passing action model

This section starts with an overview of the *layered message-passing model* for two agents, or *processes* as they are called in distributed computing. More details about this model can be found in [12]. This model is known to be equivalent to the well-studied *read/write wait-free model*, in the sense that it solves the same set of tasks. When there are only two processes involved in the computation, which is what we want to study in this article, the layered message-passing model is easier to understand. Here, we formalize this model as an action model; a more usual presentation can be found in the long version of the paper [9], along with a proof of equivalence between the two.

The layered message-passing model. Let the processes be B, W, to draw them in the pictures with colors black and white. In the *layered message-passing* model, computation is synchronous: B and W take steps at the same time. We will call each such step a *layer*. In each layer, B and W both send a message to each other, where at most one message may fail to arrive, implying that either one or two messages will be received. This is a *full information* model, in the sense that each time a process sends a message, the message consists of its local state (i.e., all the information currently known to the process), and each time it receives a message, it appends it to its own local state (remembers everything). A protocol is defined by the number N of layers the processes execute. Then, each process should produce an output value based on its state at the end of the last layer. A decision function δ specifies the output value of each process at the end of the last layer.

Given an initial state, an execution can be specified by a sequence of N symbols over the alphabet $\{\bot, B, W\}$, meaning that, if the i-th symbol in the sequence is $\bot$ then in the i-th layer both messages arrived, and if the i-th symbol is B (resp. W) then only B's message failed to arrive (resp. W) in the i-th layer. As an example, $\bot BW$ corresponds to an execution in which both processes have received each others message at layer one, then B received the message from W but W did not receive the message from B at layer two, and finally at layer three, W received the message from B but B did not receive the message from W.

For example, there are three 1-layer executions, namely $\bot$, B and W, but from the point of view of process B, there are two distinguished cases: (i) either it did not receive a message, in which case it knows for sure that the execution that occurred was W, or (ii) it did receive a message from W, in which case the execution could have been either B or $\bot$. Thus, for the black process executions B and $\bot$ are indistinguishable.

The layered message-passing model as an action model. Consider the situation where the agents $\mathrm{Ag} = \{B, W\}$ each start in an initial global state, defined by input values given to each agent. The values are local, in the sense that each agent knows its own initial value, but not necessarily the values given to other agents. The agents communicate to each other via the layered message-passing model described above. The layered message-passing action model described next is equivalent to the immediate snapshot action model of [11] in the case of two processes.

Let V^{in} be an arbitrary domain of *input values*, and take the following set of atomic propositions $\mathrm{At} = \{\mathsf{input}_a^x \mid a \in \mathrm{Ag},\, x \in V^{in}\}$. Consider a simplicial model $\mathcal{I} = \langle I, \chi, \ell \rangle$ called the *input simplicial model.* Moreover, we assume that for each vertex $v \in \mathcal{V}(I)$, corresponding to some agent $a = \chi(v)$, the labeling $\ell(v) \subseteq \mathrm{At}_a$ is a singleton, assigning to the agent a its private input value. A facet $X \in \mathcal{F}(I)$ represents a possible initial configuration, where each agent has been given an input value.

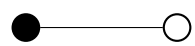
The action model $\mathcal{MP}_N = \langle T, \sim, \mathsf{pre} \rangle$ corresponding to N layers is defined as follows. Let L_N be the set of all sequences of N symbols over the alphabet $\{\bot, B, W\}$. Then, we take $T = L_N \times \mathcal{F}(I)$. An action (α, X), where $\alpha \in L_N$ and $X \in \mathcal{F}(I)$ represents a possible execution starting in the initial configuration X. We write X_a for the input value assigned to agent a in the input simplex X.

Then, $\mathsf{pre} : T \to \mathcal{L}_{\mathcal{K}}$ assigns to each $(\alpha, X) \in T$ a precondition formula $\mathsf{pre}(\alpha, X)$ which holds exactly in X (formally, we take $\mathsf{pre}(\alpha, X) = \bigwedge_{a \in \mathrm{Ag}} \mathsf{input}_a^{X_a}$). To define the indistinguishability relation $\sim_a$, we proceed by induction on N. For $N = 0$, we define $(\varnothing, X) \sim_a (\varnothing, Y)$ when $X_a = Y_a$, since process a only sees its own local state. Now assume that the indistinguishability relation of $\mathcal{MP}_N$ has been defined, we define $\sim_a$ on $\mathcal{MP}_{N+1}$ as follows. Let $\alpha, \beta \in L_N$ and $p, q \in \{\perp, B, W\}$. We define $(\alpha \cdot p, X) \sim_B (\beta \cdot q, Y)$ if either:

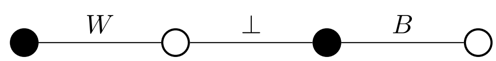
(i) $p = q = W$ and $(\alpha, X) \sim_B (\beta, Y)$, or
(ii) $p, q \in \{\perp, B\}$ and $X = Y$ and $\alpha = \beta$,

and similarly for $\sim_W$, with the role of B and W reversed. Intuitively, either (i) no message was received, and the uncertainty from the previous layers remain; or (ii) a message was received, and the process B can see the whole history, except that it does not know whether the last layer was B or $\perp$.

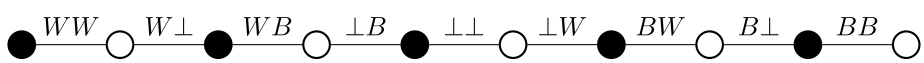
To see what the effect of this action model is, let us start with an input model $\mathcal{I}$ with only one input configuration X (input values have been omitted).



After one layer of the message passing model, we get the following model $\mathcal{I}[\mathcal{MP}_1]$:



After a second layer, we get $\mathcal{I}[\mathcal{MP}_2]$:



The remarkable property of this action model, is that it preserves the topology of the input model. This is a well-known fact in distributed computing [12], reformulated here in terms of DEL.

Theorem 1. *Let $\mathcal{I} = \langle I, \chi, \ell \rangle$ be an input model, and $\mathcal{MP}_N = \langle T, \sim, \mathsf{pre} \rangle$ be the N-layer action model. Then, the product update simplicial model $\mathcal{I}[\mathcal{MP}_N]$ is a subdivision of $\mathcal{I}$, where each edge is subdivided into 3^N edges.*

2.3 Outline of impossibility proofs

We now describe how the set up of [11] is used to prove impossibility results in distributed computing. It is closely related to the usual topological approach to distributed computability [12], except that the input complex, output complex and protocol complex are now viewed as simplicial models for epistemic logic. By interpreting epistemic logic formulas on those structures, we can understand the epistemic content of the abstract topological arguments for unsolvability. For example, when the usual topological proof would claim that consensus is not solvable because the protocol complex is connected, our DEL framework allows us to say that the reason for impossibility is that the processes did not reach common knowledge of the set of input values. This particular example, among others, is treated in depth in [11].

As in the previous section, we fix an input simplicial model $\mathcal{I} = \langle I, \chi, \ell \rangle$. A *task* for $\mathcal{I}$ is an action model $\mathcal{T} = \langle T, \sim, \mathsf{pre} \rangle$ for agents Ag, where each action $t \in T$ consists of a function $t : \mathrm{Ag} \to V^{out}$, where V^{out} is an arbitrary domain of *output values*. Such an action is interpreted as an assignment of an output value for each agent. Each such t has a precondition that is true in one or more facets of $\mathcal{I}$, interpreted as "if the input configuration is a facet in which $\mathsf{pre}(t)$ holds, and every agent $a \in \mathrm{Ag}$ decides the value $t(a)$, then this is a valid execution". The indistinguishability relation is defined as $t \sim_a t'$ when $t(a) = t'(a)$.

Definition 4. *The task $\mathcal{T}$ is solvable in the N-layer message-passing model if there exists a morphism $\delta : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{I}[\mathcal{T}]$ such that $\pi \circ \delta = \pi$, i.e., the diagram of simplicial complexes below commutes.*

$$\begin{tikzcd} & \mathcal{I}[\mathcal{MP}_N] \arrow[dl, "\pi"'] \arrow[d, dashed, "\delta"] \\ \mathcal{I} & \mathcal{I}[\mathcal{T}] \arrow[l, "\pi"] \end{tikzcd}$$

In the above definition, the two maps denoted as $\pi : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{I}$ and $\pi : \mathcal{I}[\mathcal{T}] \to \mathcal{I}$ are simply projections on the first component. The intuition behind this definition is the following. A facet X in $\mathcal{I}[\mathcal{MP}_N]$ corresponds to a pair (i, act), where $i \in \mathcal{F}(\mathcal{I})$ represents input value assignments to all agents, and $act \in \mathcal{MP}_N$ represents an action, codifying the communication exchanges that took place. The morphism δ takes X to a facet $\delta(X) = (i, t)$ of $\mathcal{I}[\mathcal{T}]$, where $t \in \mathcal{T}$ is assignment of decision values that the agents will choose in the situation X.

Moreover, $\mathsf{pre}(t)$ holds in i, meaning that t corresponds to valid decision values for input i. The commutativity of the diagram expresses the fact that both X and $\delta(X)$ correspond to the same input assignment i. Now, consider a single vertex $v \in X$ with $\chi(v) = a \in \mathrm{Ag}$. Then, agent a decides its value solely according to its knowledge in $\mathcal{I}[\mathcal{MP}_N]$: if another facet X' contains v, then $\delta(v) \in \delta(X) \cap \delta(X')$, meaning that a has to decide the same value in both situations.

To prove impossibility results, our goal is thus to show that no such map δ can exist. To do so, we rely on the following lemma, which is a reformulation in the simplicial setting of a classic result of modal logics.

Lemma 1 ([11]). *Consider simplicial models $\mathcal{M} = \langle C, \chi, \ell \rangle$ and $\mathcal{M}' = \langle C', \chi', \ell' \rangle$, and a morphism $f : \mathcal{M} \to \mathcal{M}'$. Let $X \in \mathcal{F}(C)$ be a facet of $\mathcal{M}$, a an agent, and φ a formula which does not contain negations except, possibly, in front of atomic propositions. Then, $\mathcal{M}', f(X) \models \varphi$ implies $\mathcal{M}, X \models \varphi$.*

To prove that a task $\mathcal{T}$ is not solvable in $\mathcal{MP}_N$, our usual proof method goes like this. Assume $\delta : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{I}[\mathcal{T}]$ exists, then:

1. Pick a well-chosen positive epistemic logic formula φ,
2. Show that φ is true in every world of $\mathcal{I}[\mathcal{T}]$,
3. Show that there exists a world X of $\mathcal{I}[\mathcal{MP}_N]$ where φ is false,
4. By Lemma 1, since φ is true in $\delta(X)$ then it must also be true in X, which is a contradiction with the previous point.

This kind of proof is interesting because it explains the reason why the task is not solvable. The formula φ represents some amount of knowledge which the processes must acquire in order to solve the task. If φ is given, the difficult part of the proof is usually the third point: finding a world X in the protocol complex where the processes did not manage to obtain the required amount of knowledge. The existence of this world can be proved using theorems of combinatorial topology, such as Sperner's Lemma; see [11] for such examples.

3 Equality negation task for two processes

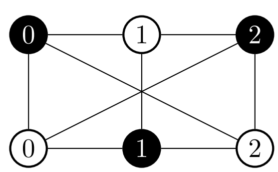
The equality negation task has been introduced in [18], and further studied in [10]. In this section, we will be interested only in the case of two processes. Each process starts with an input value in the set $\{0, 1, 2\}$, and has to irrevocably decide on a value 0 or 1, such that the decisions of the two processes are the same if and only if their input values are different. In [18], it has been proved that the equality negation task is unsolvable for two processes in a wait-free model using only registers. We reproduce this proof in the long version [9], as well as a more topological proof. In this section we analyze this task using our DEL framework and use it to prove the unsolvability of the equality negation task.

3.1 DEL analysis of the task

Let $\mathrm{Ag} = \{B, W\}$ be the two agents (or processes). In the pictures, process B will be associated to black vertices, and process W with white vertices. The atomic propositions are of the form input_p^i, for $p \in \mathrm{Ag}$ and $i \in \{0, 1, 2\}$, meaning that process p has input value i. The input model is $\mathcal{I} = \langle I, \chi, \ell \rangle$ where:

- I is the simplicial complex whose set of vertices is $\mathcal{V}(I) = \mathrm{Ag} \times \{0, 1, 2\}$, and whose facets are of the form $\{(B, i), (W, j)\}$ for all i, j.
- The coloring $\chi : \mathcal{V}(I) \to \mathrm{Ag}$ is the first projection $\chi(p, i) = p$.
- $\ell(p, i) = \{\mathsf{input}_p^i\}$.

The input model $\mathcal{I}$ is represented below. In the picture, a vertex $(p, i) \in \mathcal{V}(I)$ is represented as a vertex of color p with value i.

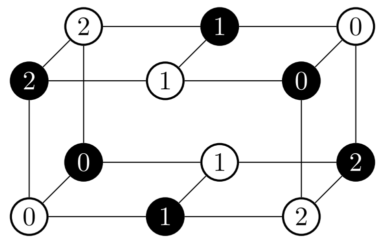


We now define the action model $\mathcal{T} = \langle T, \sim, \mathsf{pre} \rangle$ that specifies the task. Since the only possible outputs are 0 and 1, there are four possible actions: $T = \{0, 1\}^2$, where by convention the first component is the decision of B, and the second component is the decision of W. Thus, two actions $(d_B, d_W) \sim_B (d'_B, d'_W)$ in T are indistinguishable by B when $d_B = d'_B$, and similarly for W. Finally, the precondition $\mathsf{pre}(d_B, d_W)$ specifies the task as expected: if $d_B = d_W$ then

$\mathsf{pre}(d_B, d_W)$ is true exactly in the simplices of $\mathcal{I}$ which have different input values, and otherwise in all the simplices which have identical inputs.

The *output model* is obtained as the product update model $\mathcal{O} = \mathcal{I}[\mathcal{T}] = \langle O, \chi_{\mathcal{O}}, \ell_{\mathcal{O}} \rangle$. By definition, the vertices of O are of the form (p, i, E), where $(p, i) \in \mathcal{V}(I)$ is a vertex of $\mathcal{I}$, and E is an equivalence class of $\sim_p$. But note that $\sim_p$ has only two equivalence classes, depending on the decision value (0 or 1) of process p. So a vertex of O can be written as (p, i, d), meaning intuitively that process p started with input i and decided value d. The facets of O are of the form $\{(B, i, d_B), (W, j, d_W)\}$ where either $i = j$ and $d_B \neq d_W$, or $i \neq j$ and $d_B = d_W$. The coloring $\chi_{\mathcal{O}}$ and labeling $\ell_{\mathcal{O}}$ behave the same as in $\mathcal{I}$.

The output model for the equality negation task is depicted below. Decision values do not appear explicitly on the picture, but notice how the vertices are arranged as a rectangular cuboid: the vertices on the front face have decision value 0, and those on the rear face decide 1.



We want to prove that this task is not solvable when the processes communicate through N layers of our message passing model, no matter how large N is selected. Thus, we need to show that there is no morphism $\delta : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{O}$ that makes the diagram of Definition 4 commute. In Section 3.2, we will show that the general proof method described in Section 2.3 actually fails. In Section 3.3, we extend the expressivity of our logic in order to obtain an epistemic proof that δ does not exist.

3.2 Bisimulation and limits of the DEL framework

We would like to use the proof method described in Section 2.3 to find a logical obstruction showing that the morphism δ cannot exist, through a formula φ. To show that, in fact, there is no such suitable formula φ, we first need to define bisimulations for simplicial models.

Definition 5 (Bisimulation). *Let $\mathcal{M} = \langle M, \chi, \ell \rangle$ and $\mathcal{M}' = \langle M', \chi', \ell' \rangle$ be two simplicial models. A relation $R \subseteq \mathcal{F}(M) \times \mathcal{F}(M')$ is a* bisimulation *between $\mathcal{M}$ and $\mathcal{M}'$ if the following conditions hold:*

(i) If $X \, R \, X'$ then $\ell(X) = \ell'(X')$.
(ii) For all $a \in \mathrm{Ag}$, if $X \, R \, X'$ and $a \in \chi(X \cap Y)$, then there exists $Y' \in \mathcal{F}(M')$ such that $Y \, R \, Y'$ and $a \in \chi'(X' \cap Y')$.
(iii) For all $a \in \mathrm{Ag}$, if $X \, R \, X'$ and $a \in \chi'(X' \cap Y')$, then there exists $Y \in \mathcal{F}(M)$ such that $Y \, R \, Y'$ and $a \in \chi(X \cap Y)$.

When R is a bisimulation and $X \mathrel{R} X'$, we say that X and X' are bisimilar*.*

The next lemma states that two bisimilar worlds satisfy exactly the same formulae. This is a well-known fact in the context of Kripke models. The same results holds for bisimulations between simplicial models.

Lemma 2. *Let R be a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$. Then for all facets X, X' such that $X \mathrel{R} X'$, and for every epistemic logic formula φ,*

$$\mathcal{M}, X \models \varphi \qquad \textit{iff} \qquad \mathcal{M}', X' \models \varphi$$

We now come back to the equality negation task for two processes. As it turns out, there is a bisimulation between the input and output models.

Lemma 3. *Let $\mathcal{I}$ and $\mathcal{O}$ be the input and output models of the equality negation task, respectively, and let π be the projection map $\pi : \mathcal{O} \to \mathcal{I}$. The relation $R = \{(\pi(X), X) \mid X \in \mathcal{F}(\mathcal{O})\} \subseteq \mathcal{I} \times \mathcal{O}$ is a bisimulation between $\mathcal{I}$ and $\mathcal{O}$.*

Proof. The first condition of Definition 5 is trivially fulfilled.

Let us check that condition (ii) is verified. Let X and X' be facets of $\mathcal{I}$ and $\mathcal{O}$ respectively, such that $X \mathrel{R} X'$. Thus, we have $X = \{(B, i), (W, j)\}$ and $X' = \{(B, i, d_B), (W, j, d_W)\}$, for some i, j, d_B, d_W. Now let $a \in \mathrm{Ag}$ (w.l.o.g., let us pick $a = B$), and assume that there is some $Y \in \mathcal{F}(\mathcal{I})$ such that $B \in \chi(X \cap Y)$. So, Y can be written as $Y = \{(B, i), (W, j')\}$ for some j'. We now need to find a facet Y' of $\mathcal{O}$ that shares a B-colored vertex with X', and whose projection $\pi(Y')$ is Y. Thus, Y' should be of the form $Y' = \{(B, i, d_B), (W, j', d'_W)\}$, for some d'_W, such that $i = j' \iff d_B \neq d'_W$. But whatever the values of i, j', d_B are, we can always choose a suitable d'_W. This concludes the proof.

The third condition (iii) is checked similarly. □

We can finally use Lemma 2 to show that no formula φ will allow us to prove the unsolvability of the equality negation task.

Lemma 4. *For the equality negation task, let X be a facet of $\mathcal{I}[\mathcal{MP}_N]$ and let Y be a facet of $\mathcal{O}$ such that $\pi(X) = \pi(Y)$. Then for every positive formula φ we have the following: if $\mathcal{O}, Y \models \varphi$ then $\mathcal{I}[\mathcal{MP}_N], X \models \varphi$.*

Proof. Let φ be a positive formula and assume $\mathcal{O}, Y \models \varphi$. Since we have shown in Lemma 3 that $\pi(Y)$ and Y are bisimilar, by Lemma 2, we have $\mathcal{I}, \pi(Y) \models \varphi$. Since $\pi(Y) = \pi(X)$, by Lemma 1 we obtain $\mathcal{I}[\mathcal{MP}_N], X \models \varphi$. □

In the above lemma, the world Y should be thought of as a candidate for $\delta(X)$. The condition $\pi(X) = \pi(Y)$ comes from the commutative diagram of Definition 4. Thus, Lemma 4 says that we will never find a formula φ which is true in $\delta(X)$ but false in X.

Remark. As previously discussed, Lemma 4 does not apply to consensus, since we know that there exists a formula proving its unsolvability. The reason is that the projection mapping $\pi : \mathcal{O} \to \mathcal{I}$ in consensus does not induce a bisimulation.

Here we show that condition (ii) of Definition 5 does not hold. Namely, if $X = \{(B,0),(W,1)\}$ and $X' = \{(B,0,1),(W,1,1)\}$ and $Y = \{(B,0),(W,0)\}$, then by definition of consensus there cannot exist a facet Y' with $Y \; R \; Y'$ and $B \in \chi'(X' \cap Y')$. Such a facet would have the form $Y' = \{(B,0,1),(W,0,d)\}$, for a $d \in \{0,1\}$, which is not a valid world in the output model of consensus for any decision d.

3.3 Extended DEL

In Section 3.2, we have shown that no epistemic logic formula is able to express the reason why the equality negation task is not solvable. This seems to indicate that our logic is too weak: indeed, because of the product update model construction that we use, we are only allowed to write formulas about the inputs and what the processes know about each other's inputs. But the specification of the task is very much about the outputs too! If we allow ourselves to use atomic propositions of the form decide_p^d, with the intended meaning that process p decides value d, a good candidate for the formula φ seems to be:

$$\varphi \; = \; \bigwedge_{p,i,d} \mathsf{input}_p^i \wedge \mathsf{decide}_p^d \implies \Big((\mathsf{input}_{\bar{p}}^i \wedge \mathsf{decide}_{\bar{p}}^{\bar{d}}) \vee (\mathsf{input}_{\bar{p}}^{\bar{i}} \wedge \mathsf{decide}_{\bar{p}}^{d}) \Big)$$

where $\bar{p}$, $\bar{i}$, $\bar{d}$ denote values different from p, i, d, respectively. Note that $\bar{p}$ and $\bar{d}$ are uniquely defined (since there are only two processes and two decision values), but for $\bar{i}$, there are two possible inputs different from i. So, for example, $\mathsf{input}_p^{\bar{0}}$ is actually a shortcut for $\mathsf{input}_p^1 \vee \mathsf{input}_p^2$.

This formula simply expresses the specification of the task: if process p has input i and decides d, then the other process should either have the same input and decide differently, or have a different input and decide the same. Then hopefully φ would be true in every world of the output complex, but would fail somewhere in the protocol complex $\mathcal{I}[\mathcal{MP}_N]$, meaning that the N-layer message-passing model is not powerful enough to obtain this knowledge.

To be able to express such a formula, we first need to enrich our models by saying in which worlds the atomic propositions decide_p^d are true or false. Let $\widehat{\mathrm{At}} = \mathrm{At} \cup \{\mathsf{decide}_p^d \mid p \in \mathrm{Ag}, d \in \{0,1\}\}$ be the new set of atomic propositions. The definition of the *extended product update model* $\widehat{\mathcal{I}[\mathcal{T}]} = \widehat{\mathcal{O}}$ is straightforward:

- Its vertices are of the form (p,i,d) with $p \in \mathrm{Ag}$, $i \in \{0,1,2\}$ and $d \in \{0,1\}$. The facets are $\{(B,i,d_B),(W,j,d_W)\}$ where $i = j \iff d_B \neq d_W$.
- The coloring map is $\chi_{\widehat{\mathcal{O}}}(p,i,d) = p$.
- The atomic propositions labeling is $\ell_{\widehat{\mathcal{O}}}(p,i,d) = \{\mathsf{input}_p^i, \mathsf{decide}_p^d\}$.

Thus, this is almost the same model as the one of Section 3.1, but we have added some annotations to say where the decide_p^d atomic propositions are true. It is easily checked that the formula φ is true in every world of $\widehat{\mathcal{O}}$.

Now, we would also like the formula φ to make sense in the protocol complex $\mathcal{I}[\mathcal{MP}_N]$, but it does not seem to have any information about decision values.

It only describes the input values, and which execution has occurred. But it is precisely the role of the simplicial map $\delta : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{O}$ to assign decision values to each world of $\mathcal{I}[\mathcal{MP}_N]$. Thus, given such a map δ, we can lift it to a map $\widehat{\delta} : \mathcal{I}[\widehat{\mathcal{MP}_N}] \to \widehat{\mathcal{O}}$ as the following lemma states.

Lemma 5. *Let $\mathcal{M} = \langle M, \chi, \ell \rangle$ be a simplicial model over the set of agents* Ag *and atomic propositions* At*, and let $\delta : \mathcal{M} \to \mathcal{O}$ be a morphism of simplicial models. Then there is a unique model $\widehat{\mathcal{M}} = \langle M, \chi, \widehat{\ell} \rangle$ over $\widehat{\mathrm{At}}$, where $\widehat{\ell}$ agrees with ℓ on* At*, such that $\widehat{\delta} : \widehat{\mathcal{M}} \to \widehat{\mathcal{O}}$ is still a morphism of simplicial models.*

Proof. All we have to do is label the worlds of $\mathcal{M}$ with the decide_p^d atomic propositions, such that δ is a morphism of simplicial models. Thus, we define $\widehat{\ell} : M \to \mathscr{P}(\widehat{\mathrm{At}})$ as $\widehat{\ell}(m) = \ell(m) \cup \{\mathsf{decide}_p^d\}$, where $\delta(m) = (p, i, d) \in O$. Then δ is still a chromatic simplicial map (since we did not change the underlying complexes nor their colors), and moreover we have $\widehat{\ell}(m) = \ell_{\widehat{\mathcal{O}}}(\delta(m))$ for all m. The model $\widehat{\mathcal{M}}$ is unique since any other choice of $\widehat{\ell}(m)$ would have broken this last condition, so δ would not be a morphism of simplicial models. □

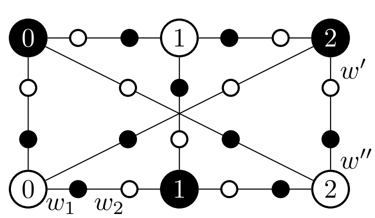
We can finally prove that the equality negation task is not solvable:

Theorem 2. *The equality negation task for two processes is not solvable in the N-layer message-passing model.*

Proof. Let us assume by contradiction that the task is solvable, i.e., by Definition 4, there exists a morphism of simplicial models $\delta : \mathcal{I}[\mathcal{MP}_N] \to \mathcal{O}$ that makes the diagram commute. By Lemma 5, we can lift δ to a morphism $\widehat{\delta} : \mathcal{I}[\widehat{\mathcal{MP}_N}] \to \widehat{\mathcal{O}}$ between the extended models. As we remarked earlier, the formula φ is true in every world of $\widehat{\mathcal{O}}$. Therefore, it also has to be true in every world of $\mathcal{I}[\widehat{\mathcal{MP}_N}]$. Indeed, for any world w, since $\widehat{\mathcal{O}}, \delta(w) \models \varphi$, and δ is a morphism, by Lemma 1, we must have $\mathcal{I}[\widehat{\mathcal{MP}_N}], w \models \varphi$. We will now derive a contradiction from this fact.

$$\begin{tikzcd} & \mathcal{I}[\mathcal{MP}_N] \arrow[dl, "\pi"'] \arrow[d, "\delta"] & \mathcal{I}[\widehat{\mathcal{MP}_N}] \arrow[d, "\widehat{\delta}"] \\ \mathcal{I} & \mathcal{O} \arrow[l, "\pi"] & \widehat{\mathcal{O}} \end{tikzcd}$$

Recall that the protocol complex $\mathcal{I}[\mathcal{MP}_N]$ is just a subdivision of the input complex $\mathcal{I}$, as depicted below. (For simplicity, some input values have been omitted in the vertices on a subdivided edge; it is the same input as the extremity of the edge which has the same color. Also, the picture shows only one subdivision, but our reasoning is unrestricted and it applies to any number of layers N.)



Let us start in some world w_1 on the $(W,0)-(B,1)$ edge. In the world w_1, the two processes have different inputs. Since in $\mathcal{I}[\widehat{\mathcal{MP}_N}]$, the formula φ is true in w_1, the decision values have to be the same. Without loss of generality, let us assume that in w_1, both processes decide 0.

We then look at the next world w_2, which shares a black vertex with w_1. Since the inputs are still 0 and 1, and φ is true, and we assumed that process B decides 0, then the white vertex of w_2 also has to decide 0.

We iterate this reasoning along the $(W,0)-(B,1)$ edge, then along the $(B,1)-(W,2)$ edge, and along the $(W,0)-(B,2)$ edge: all the vertices on these edges must have the same decision value 0. Thus, on the picture, the top right $(B,2)$ corner has to decide 0, as well as the bottom right $(W,2)$ corner.

Now in the world w', the two input values are equal, so the processes should decide differently. Since the black vertex decides 0, the white vertex must have decision value 1. If we keep going along the rightmost edge, the decision values must alternate: all the black vertices must decide 0, and the white ones decide 1. Finally, we reach the world w'', where both decision values are 0, whereas the inputs are both 2. So the formula φ is false in w'', which is a contradiction. $\square$

It is interesting to compare the epistemic formula φ that we used in this paper to prove the unsolvability of equality negation, with the one (let us call it ψ) that was used in [11] to prove the impossibility of solving consensus. In the case of consensus, we did not need the "Extended DEL" framework. The formula ψ was simply saying that the processes have common knowledge of the input values. This formula is quite informative: it tells us that the main goal of the consensus task is to achieve common knowledge. On the other hand, the formula φ is less informative: it is simply stating the specification of the equality negation task. It does not even seem to be talking about knowledge, since there are no K or C operators in the formula. In fact, the epistemic content of φ is hidden in the decide_p^d atomic propositions. Indeed, their semantics in $\mathcal{I}[\widehat{\mathcal{MP}_N}]$ is referring to the decision map δ, which assigns a decision value d to each vertex of $\mathcal{I}[\mathcal{MP}_N]$. The fact that we assign decisions to vertices means that each process must decide its output *solely according to its knowledge.*

Despite the fact that it produces less informative formulas, the "Extended DEL" proof method has two major benefits. First, it seems to be able to prove *any* impossibility result. Indeed, let $\mathcal{T} = \langle T, \sim, \mathsf{pre}\rangle$ be a task action model, on the input model $\mathcal{I}$, and let $\mathcal{P}$ be a protocol action model. Remember that the elements of T are functions $t : \mathrm{Ag} \to V^{out}$ assigning a decision value to each agent. Let φ denote the following formula:

$$\varphi \;=\; \bigwedge_{X\in\mathcal{F}(\mathcal{I})} \left(\bigwedge_{p\in\mathrm{Ag}} \mathsf{input}_p^{X(p)} \implies \bigvee_{\substack{t\in T\\ \mathcal{I},X\models \mathsf{pre}(t)}} \bigwedge_{p\in\mathrm{Ag}} \mathsf{decide}_p^{t(p)} \right) \tag{1}$$

where $X(p)$ denotes the input value of process p in the input simplex X. Then we get the following Theorem (whose proof is in the long version [9]).

Theorem 3. *The task $\mathcal{T}$ is solvable in the protocol $\mathcal{P}$ if and only if there exists an extension $\widehat{\mathcal{I}[\mathcal{P}]}$ of $\mathcal{I}[\mathcal{P}]$ (assigning a single decision value to each vertex of $\mathcal{I}[\mathcal{P}]$) such that φ from (1) is true in every world of $\widehat{\mathcal{I}[\mathcal{P}]}$.*

This theorem implies that the situation of Section 3.2 cannot happen with the "Extended DEL" approach: if the task is not solvable, there necessarily exists a world X of $\widehat{\mathcal{I}[\mathcal{P}]}$ where the formula fails. Of course, finding such a world is usually the hard part of an impossibility proof, but at least we know it exists. In fact, in the particular case of read/write protocols (or, equivalently, layered message-passing), the solvability of tasks is known to be undecidable when there are more than three processes [8, 14]. Thus, according to our Theorem, given a formula φ, the problem of deciding whether there exists a number of layers N and an extension of $\mathcal{I}[\mathcal{MP}_N]$ which validates the formula, is also undecidable.

The second benefit of the "Extended DEL" framework is that it gives us a way of using epistemic logic as a specification language for tasks. Notice that in Theorem 3, we characterized the solvability of a task without referring to $\mathcal{T}$ itself: the formula φ contains all the information of $\mathcal{T}$. Thus, instead of relying on the commutative diagram of Definition 4, we can specify a task directly as a logical formula. One could decide to pick a more informative formula, with an interesting epistemic content, and study the solvability of this "task".

4 Conclusion

The equality negation task is known to be unsolvable in the wait-free read/write model. In this paper, we gave a new proof of this result, using the simplicial complex semantics of DEL that we proposed in [11]. There are two purposes of doing this. First, the logical formula witnessing the unsolvability of a task usually helps us understand the epistemic content of this task. Unfortunately, as it turns out, the logical formula that we obtained in the end is less informative than we hoped. Secondly, this is a nice case study to test the limits of our DEL framework. Indeed, we proved in Section 3.2 that the basic language of DEL, where formulas are only allowed to talk about input values, is too weak to express the reason why the task is not solvable. To fix this issue, we introduced a way to extend our logical language in order to have more expressive formulas.

Acknowledgements

The authors were supported by DGA project "Validation of Autonomous Drones and Swarms of Drones" and the academic chair "Complex Systems Engineering" of Ecole Polytechnique-ENSTA-Télécom-Thalès-Dassault-Naval Group-DGA-FX-FDO-Fondation ParisTech, by the UNAM-PAPIIT project IN109917, by the France-Mexico Binational SEP-CONACYT-ANUIES-ECOS grant M12M01, by the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme under grant agreement No 787367 (PaVeS), as well as by the Austrian Science Fund (FWF) through Doctoral College LogiCS (W1255-N23).

References

1. Attiya, H., Welch, J.: Distributed Computing: Fundamentals, Simulations and Advanced Topics. John Wiley & Sons, Inc., USA (2004)
2. Baltag, A., Moss, L., Solecki, S.: The logic of common knowledge, public announcements, and private suspicions. In: TARK VII. pp. 43–56 (1998). https://doi.org/10.1007/978-3-319-20451-2_38
3. Baltag, A., Renne, B.: Dynamic epistemic logic. In: The Stanford Encyclopedia of Philosophy, see https://plato.stanford.edu/archives/win2016/entries/dynamic-epistemic/. Metaphysics Research Lab, Stanford University (2016)
4. Biran, O., Moran, S., Zaks, S.: A combinatorial characterization of the distributed 1-solvable tasks. J. Algorithms **11**(3), 420–440 (1990). https://doi.org/10.1016/0196-6774(90)90020-F
5. Chor, B., Israeli, A., Li, M.: On processor coordination using asynchronous hardware. In: Proceedings of the Sixth Annual ACM Symposium on Principles of Distributed Computing. pp. 86–97. PODC '87, ACM, New York, NY, USA (1987). https://doi.org/10.1145/41840.41848
6. Ditmarsch, H.v., van der Hoek, W., Kooi, B.: Dynamic Epistemic Logic. Springer (2007). https://doi.org/10.1007/978-1-4020-5839-4
7. Fischer, M.J., Lynch, N.A., Paterson, M.: Impossibility of distributed consensus with one faulty process. J. ACM **32**(2), 374–382 (1985). https://doi.org/10.1145/3149.214121
8. Gafni, E., Koutsoupias, E.: Three-processor tasks are undecidable. SIAM J. Comput. **28**(3), 970–983 (1999)
9. Goubault, E., Lazić, M., Ledent, J., Rajsbaum, S.: A dynamic epistemic logic analysis of the equality negation task. CoRR **abs/1909.03263** (2019), http://arxiv.org/abs/1909.03263
10. Goubault, E., Lazić, M., Ledent, J., Rajsbaum, S.: Wait-free solvability of equality negation tasks. In: 33rd International Symposium on Distributed Computing, DISC 2019, Budapest, Hungary, October 14-18, 2019 (to appear)
11. Goubault, É., Ledent, J., Rajsbaum, S.: A simplicial complex model for dynamic epistemic logic to study distributed task computability. In: Proceedings Ninth International Symposium on Games, Automata, Logics, and Formal Verification, GandALF 2018, Saarbrücken, Germany, 26-28th September 2018. pp. 73–87 (2018). https://doi.org/10.4204/EPTCS.277.6
12. Herlihy, M., Kozlov, D., Rajsbaum, S.: Distributed Computing Through Combinatorial Topology. Elsevier-Morgan Kaufmann (2013). https://doi.org/10.1016/C2011-0-07032-1
13. Herlihy, M.: Wait-free synchronization. ACM Trans. Program. Lang. Syst. **13**(1), 124–149 (Jan 1991). https://doi.org/10.1145/114005.102808
14. Herlihy, M., Rajsbaum, S.: The decidability of distributed decision tasks (extended abstract). In: Proceedings of the Twenty-Ninth Annual ACM Symposium on the Theory of Computing (STOC), El Paso, Texas, USA, May 4-6, 1997. pp. 589–598 (1997)
15. Herlihy, M., Shavit, N.: The topological structure of asynchronous computability. J. ACM **46**(6), 858–923 (1999). https://doi.org/10.1145/331524.331529
16. Jayanti, P.: On the robustness of herlihy's hierarchy. In: Proceedings of the Twelfth Annual ACM Symposium on Principles of Distributed Computing. pp. 145–157. PODC '93, ACM, New York, NY, USA (1993). https://doi.org/10.1145/164051.164070

17. Kozlov, D.: Combinatorial Algebraic Topology. Springer (2007). https://doi.org/10.1007/978-3-540-71962-5
18. Lo, W., Hadzilacos, V.: All of us are smarter than any of us: Nondeterministic wait-free hierarchies are not robust. SIAM J. Comput. **30**(3), 689–728 (2000). https://doi.org/10.1137/S0097539798335766
19. Loui, M.C., Abu-Amara, H.H.: Memory requirements for agreement among unreliable asynchronous processes. In: Advances in Computing research. pp. 163–183. JAI Press, Greenwich, CT (1987)
20. Lynch, N.A.: Distributed Algorithms. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA (1996)