Formalizing and checking Multilevel Consistency*

Ahmed Bouajjani[1], Constantin Enea[1], Madhavan Mukund[2,3], Gautham Shenoy R[2], and S P Suresh[2,3]

[1] Université Paris Diderot, France {abou,cenea}@irif.fr
[2] Chennai Mathematical Institute, India {madhavan,gautshen,spsuresh}@cmi.ac.in
[3] CNRS UMI 2000 ReLaX

Abstract. Developers of distributed data-stores must in general trade consistency for performance and availability. Such systems may in fact implement weak consistency models, e.g., causal consistency or eventual consistency, corresponding to different costs and guarantees to the clients. In this work, we consider the case of distributed systems that offer not just one level of consistency but multiple levels of consistency to the clients. This corresponds to many practical situations. For instance, popular data-stores such as Amazon DynamoDB and Apache's Cassandra allow applications to tag each query within the same session with a separate consistency level. Other examples are data-store implementations with incremental correctness guarantees allowing an application to obtain a sequence of responses to a query corresponding to increasingly strong levels of consistency. In this paper, we provide a formal framework for the specification of multilevel consistency, and we address the problem of checking the conformance of a computation with respect to such a model. We provide a principled algorithmic approach for solving this problem and apply it to several instances of models with multilevel consistency.

1 Introduction

To achieve availability and scalability, modern data-stores (key-value stores) rely on optimistic replication, allowing multiple clients to issue operations on shared data on a number of replicas, which communicate changes to each other using message passing. One benefit of such architectures is that the replicas remain locally available to clients even when network connections fail. Unfortunately, the famous CAP theorem [10] shows that such high Availability and tolerance to network Partitions are incompatible with strong Consistency, i.e., the illusion of a single centralized replica handling all operations. For this reason, modern replicated data-stores often provide weaker forms of consistency such as eventual consistency [18] or causal consistency [14], which have been formalized only recently [3, 4, 6, 17].

* Partially supported by CEFIPRA DST-Inria-CNRS Project 2014-1, AVeCSo.

Programming applications on top of weakly-consistent data-stores is difficult. Some form of synchronization is most often unavoidable in order to preserve correctness. Therefore, popular data-stores such as Amazon DynamoDB and Apache's Cassandra provide different levels of consistencies, ranging from weaker forms to strong consistency. Applications can tag queries to the data-store with a suitable level of consistency depending on their needs.

Implementations of large-scale data-stores are difficult to build and test. For instance, they must account for partial failures, where some components or the network can fail and produce incomplete results. Ensuring fault-tolerance relies on intricate protocols which are difficult to design and reason about. The black-box testing framework Jepsen [4] found a remarkably large number of subtle problems in many production distributed data-stores.

Testing a data-store raises two issues: (1) deriving a suitable set of testing scenarios, e.g., faults to inject into the system and the set of operations to be executed, and (2) efficient algorithms for checking whether a given execution satisfies the considered consistency models. The Jepsen framework shows that the first issue can be solved using randomization, e.g., introducing faults at random and choosing the operations randomly. The effectiveness of this solution has been proved formally in recent work [16]. The second issue is dependent on a suitable formalization of the consistency models.

In this work, we consider the problem of specifying data-stores which provide multiple levels of consistency and derive algorithms to check whether a given execution adheres to such a multilevel consistency specification.

Concerning the formalization, we build on the specification framework in [6] which formalizes consistency models using two auxiliary relations, a *visibility* relation defining for each operation (read or write of a key) the set of operations it observes, and an *arbitration order* defining the order in which operations should be viewed by different replicas. An execution is defined to satisfy a consistency model if we can find a visibility relation and an arbitration order that obey certain axioms. For the case of a data-store providing multiple levels of consistency, we consider multiple visibility relations and arbitration orders, one for each level of consistency. Then, we consider a set of axioms which specifies each consistency level in isolation, and also, how visibility relations and arbitration orders of different consistency levels are related.

Based on this formalization, we investigate the problem of checking whether a given execution satisfies a certain multilevel consistency specification. In general, this problem is known to be NP-complete [3]. However, we show that for some particular set of executions, where each value is written at most once (to some key), this problem becomes polynomial time for many practically-interesting multilevel consistency specifications. Since practical data-store implementations are data-independent [19], i.e., their behavior doesn't depend on the concrete values read or written in the transactions, any potential buggy behavior can be exposed in such executions. This complexity result uses the idea of *bad patterns* introduced in [3] for the case of causal consistency. Intuitively, a bad pattern can

[4] Available at `http://jepsen.io`

be seen as a set of operations occurring (within an execution) in some particular order corresponding to a consistency violation. In this paper, we provide a *systematic methodology* for deriving bad patterns characterizing a wide range of consistency models and combinations thereof.

Combined, these contributions form an effective algorithmic framework for the verification of modern data-stores providing multiple levels of consistency. To the best of our knowledge, we are the first to investigate the asymptotic complexity for such a wide class of consistency models and their combinations, despite their prevalence in practice.

The paper is organized as follows. We begin by illustrating the idea of multilevel consistency through some real-life examples. In Section 3, we present a formal model for specifying and reasoning about multilevel consistency. Section 4 describes algorithms for verifying multilevel cnonsistency. We conclude with a discussion of related work. Some details and proofs are presented in an Appendix.

2 Multilevel consistency in the wild

In this section we present some instances of multilevel consistency found in the wild. For the purpose of this paper, we restrict our attention to distributed read-write key-value data-stores (henceforth referred to as Read-write stores), consisting of unique memory locations addressed by *keys* or *variables*. We use *keys* and *variables* interchangeably in this work. The contents of these memory locations come from a domain, called *values*.

The read-write data-store provides two APIs to access and modify the contents of a particular memory location. The API to read the content of a particular memory location is typically named *Read* or *Get*, and the API to store a value into a particular memory location is typically named *Write* or *Put*. In this paper, we refer to these two methods as Read and Write respectively. The Read method does not update the state of the data-store but only reveals part of the state to the application session which invokes the method. The Write method on the other hand ends up modifying the state of the data-store.

Typically, applications read some location of the data-store, perform some local computation and write some value back to the data-store. The related sequence of read and write operations performed by the application is called a *session*.

Applications can expect some sort of consistency guarantee from the data-store in terms of how *fresh* or *stale* the data value is that they read from the data-store. They can also seek some guarantees pertaining to monotonicity of the results that are presented to them. These guarantees provided by the data-store to the applications are called *consistency criteria* in the literature. Some of the popular consistency criteria include:

- **Read-Your-Writes**: The effects of prior operations in the session will be visible to the later operations in the same session.

– **Monotonic Reads**: Once the effect of some operation becomes visible within a session, it remains visible to all the subsequent operations in that session.
– **Monotonic Writes**: If the effect of a remote operation is visible in a session, then the effects of all prior operations in the session of the remote operation will also be visible.
– **Causal consistency**: Effects of prior operations in a session are always visible to later operations. Further, if the effect of an operation is visible to another operation, then every operation that has seen the effects of the latter would have seen the effects of the former.
– **Sequential Consistency**: Effects of prior operations in a session are always visible to later operations. And for every pair of operations, either the effect of one is visible the other or vice-versa.

While most of the existing literature on testing the behaviour of read-write stores focus on testing the correctness with respect to specific consistency criteria [3, 4, 9], there are cases where data-stores such as DynamoDB and Cassandra offer to applications the choice of specifying the consistency level per read-operation [7]. There are distributed data-store libraries that allow consistency rationing [13] and also allow incremental consistency guarantees for the read operations [12]. Further, there are distributed data-store libraries that allow an application to *upgrade* the consistency level offered by the underlying data-store to a stronger one [1]. In each of these cases we need to reason about the correctness of the behaviour of the data-store with respect to more than one consistency criterion.

We now look at some examples of multilevel consistency in the real world. We assume that the Read and Write APIs are as follows:

– Write(x, val) : Updates the content of the memory location addressed by the key/variable x, with the value val.
– Read$(x, val, level)$: The content of the memory location whose key is x, is val with respect to the consistency level $level$.

Read-Write Stores with strong and weak reads

In case of DynamoDB, the data-store allows the application a choice of two consistency levels for every query that it makes. If strong consistency is chosen the query will complete only after all the replicas have been consulted and a consensus has been arrived at. If the weaker eventual consistency is chosen, then the query will complete after consulting a subset of the replicas. In case of Cassandra the data-store allows the application a more fine grained choice of consistency levels, such as `ANY`, `ONE`, `QUORUM`, `ALL`. It achieves this by ensuring that when the Read is made with `ANY`, the return value is provided by consulting any correct replica of the data store. Similarly, if the Read operation is submitted with `ONE`, then the return value is provided by consulting a replica that is known to contain at least one value for that key. On the other hand, if the Read is made

with `QUORUM`, the data-store returns the value after consulting majority of the replicas. Finally, if Read is made with `ALL`, then all the replicas are consulted before returning the response. Clearly, `ANY` is the weakest consistency criterion while `ALL` is the strongest consistency criterion. In general, a data-store offers responses pertaining to different consistency criteria by consulting the required subset of replicas to answer the query.

Typically a read operation under the stronger consistency criterion will take more time, since it might have to wait for all the operations to be visible, or run a consensus protocol before returning the result. In certain cases, applications may be satisfied with Read operations that return values that are correct with respec to some weaker consistency criterion. Consider a web-application that displays the available seats in a movie theater. The application can choose to read the available seats based on a weaker consistency criterion, since:

- The number of users attempting to book the seats will be more than the seats available. Waiting for a consensus or a quorum can slow down the reads for everyone. So a quicker response is desirable.
- There is a lag between the time the users gets to see the available seats and the time when the user decides to book particular seats. Since concurrent bookings are ongoing, the data displayed can anyway become stale by the time the user books the seat.
- Users can change their minds before finally settling on a set of seats, and paying for it.

Thus, the web-application can opt for a read satisfying a weaker consistency criterion while allowing the user to pick a seat, and then perform a read satisfying the stronger consistency criterion only when the user pays for it.

Consider the example in Figure 1 where a write is written to only one replica. For each session, there is a (potentially different) designated replica from which the responses to the weak reads are returned. The strong reads correspond to `ALL`.

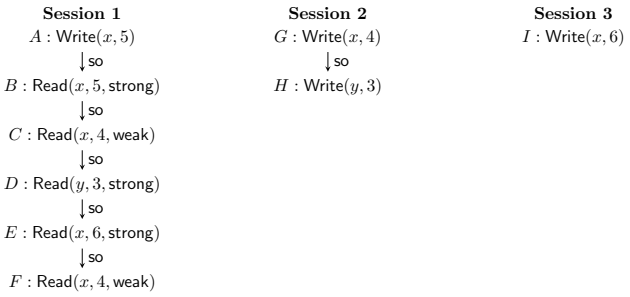
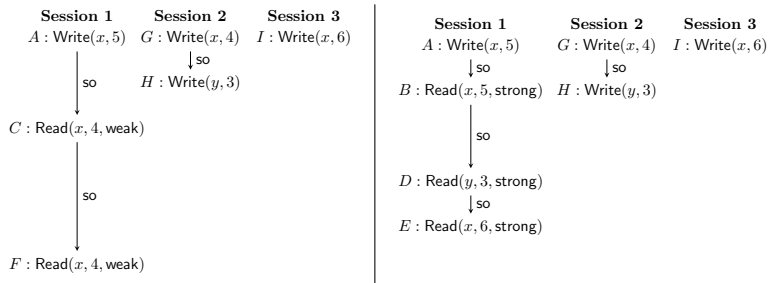


Fig. 1: An example of a read-write store behaviour with strong and weak reads

It can be seen that the strong reads correspond to sequential consistency while the weak reads correspond to monotonic reads consistency. The fragment consisting of all the writes and the weak reads should be correct with respect to monotonic reads. The fragment consisting of all the writes and the strong reads should be correct with respect to sequential consistency.



(a) Weak Fragment from Figure 1 (b) Strong Fragment from Figure 1

Fig. 2: Strong and Weak fragments of the hybrid behaviour

In the example in Figure 1, the weaker fragment can be seen in Figure 2(a). This fragment is correct with respect to monotonic reads, once the write G is visible at session 1 to the read C, it remains visible throughout the session. The write I is not visible to any of the other sessions yet.

The stronger fragment is represented in Figure 2(b). This is correct with respect to sequential consistency, as we can assume that the order of the operations obtained by consensus is $A \to B \to G \to H \to I \to D \to E$.

However, note that since the strong reads correspond to the level `ALL` where all the replicas have seen the prior writes and have agreed on the order of the concurrent writes, it behooves a weak read following a strong read to take into consideration the effects seen by the earlier strong read. Thus the data-store imposes an additional constraint that once a write is visible to a strong read in a session, it is visible to all the subsequent weak reads in that session. This ensures that the weaker reads do incorporate the prior results seen by the session.

With this additional constraint, we can no longer explain the read operation F, since the effects of writes G and I are both visible at read F. The strong consistency criteria has already guaranteed that write I has happened after write G, thereby effectively overwriting the value 4 with the value 6. Hence this behaviour is incorrect in the multilevel setting.

Bolt-on consistency

In [1], the authors provide a way of strengthening the consistency provided by a weakly consistent data-store by making visible to the application the effects of only that subset of operations that is causally complete.

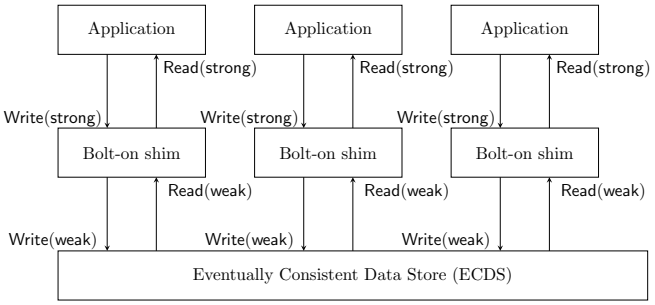


Fig. 3: Bolt-on architecture

In this framework, an application that expects a stronger consistency criterion such as causal consistency, but has access to an Eventually Consistent Data-Store (ECDS) that only offers a weak consistency guarantee such as read-your-writes can use the services of a bolt-on shim as an intermediate, which will upgrade the weak consistency offered by the ECDS to causal consistency. The architecture is shown in Figure 3.

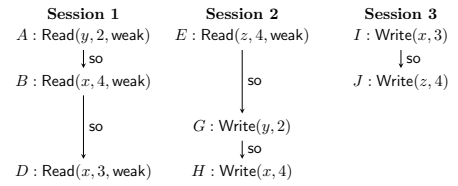
The bolt-on shim has its own local memory and a buffer. Each write made by the application is stored by the shim in its local memory. The shim then makes the same write on the ECDS, with some additional metadata to track causality across operations. From time to time, the bolt-on shim reads from the ECDS and buffers the values read. When it has read enough writes that are causally complete, it updates the local memory with these writes. Whenever the application reads a value, the value is returned from the local memory. Thus, the application sees only the causally complete fragment of all the operations seen by the shim.

As before, we model this behaviour with two kinds of reads, strong and weak. The write made by the application to the shim, and the same write forwarded by the shim to the ECDS are modelled as a single write operation.

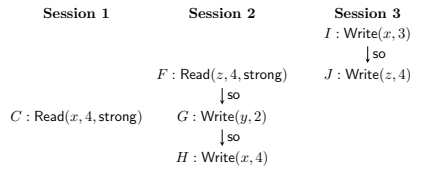
In Figure 4, we have an example of a behaviour in a Bolt-On setting where the strong consistency criterion is causal consistency and the weak consistency criterion is monotonic reads.

Session 1	**Session 2**	**Session 3**
A : Read$(y, 2,$ weak$)$	E : Read$(z, 4,$ weak$)$	I : Write$(x, 3)$
$\downarrow$so	$\downarrow$so	$\downarrow$so
B : Read$(x, 4,$ weak$)$	F : Read$(z, 4,$ strong$)$	J : Write$(z, 4)$
$\downarrow$so	$\downarrow$so	
C : Read$(x, 4,$ strong$)$	G : Write$(y, 2)$	
$\downarrow$so	$\downarrow$so	
D : Read$(x, 3,$ weak$)$	H : Write$(x, 4)$	

Fig. 4: An example of a Bolton history with strong and weak reads



(a) Weak Fragment of Bolt-on history from Fig. 4



(b) Strong Fragment of Bolt-on history from Fig. 4

Fig. 5: Weak and strong fragments of Bolt-on

The fragment containing all the write operations and the weak reads is termed the weak fragment (Figure 5(a)). This fragment is correct with respect to monotonic reads, since the only remote write visible to Session 2, is the write J, whose effects continue to remain visible throughout the session. In case of Session 1, the effect of the write G is visible to A, and that of H is visible from the read B onwards. At the read D, both the writes H and I are visible. Since the writes H and I are concurrent, we can imagine that the data-store has arbitrated H before I in the weak arbitration order, thus justifying the correctness of this behaviour with respect to monotonic reads.

The strong fragment from Figure 5(b) is correct with respect to causal consistency as we can assume that the operations have been performed in the following order: $I \rightarrow J \rightarrow F \rightarrow G \rightarrow H \rightarrow C$.

In case of bolt-on consistency, the additional constraint requires that the set of writes visible to a strong-read be a subset of the set of writes visible to prior weaker reads in the session. This is because the shim is only supposed to make visible the causally complete fragment to the application through strong reads.

In the presence of this bolt-on constraint, in the example in Figure 4, since writes G, H, I and J are visible to strong read C by the requirement of causal consistency, the bolt-on constraint will insist that these writes be visible to the weaker reads in Session 1 prior to C. Further since the weaker consistency criterion is monotonic reads, no matter which prior weak read operation these reads were first visible to, they will all be visible at read B.

Thus, since both the writes $H : \mathsf{Write}(x, 4)$ and $I : \mathsf{Write}(x, 3)$ are visible at B, and the return value of the read at B is 4, it has to be the case that the eventually consistent database has arbitrated I before H. Owing to monotonic reads, the effects of both H and I are again visible to the read D. Since I has been arbitrated before H, the read at D should have returned 4. However, from the history the weak read at D returns 3, which is incorrect.

In both the cases of multilevel consistency that we have considered in this section, we can see that the presence of another consistency criterion can impose additional constraints on the choice of the visibility and arbitration relations chosen to explain the correctness of the history. In the next section, we will

provide the formal framework for modelling behaviours of read-write data-stores with multiple consistency levels.

3 Formalizing Multilevel Consistency

We now provide the formal definitions for modeling the behaviours of read-write stores. These definitions are extensions of the formal framework provided in [5].

Each operation submitted to the data-store by the application is either a Read or a Write operation with the following signature:

- $\mathsf{Write}(x, val)$: Updates the content of the memory location addressed by the key/variable x, with the value val.
- $\mathsf{Read}(x, val, level)$: The content of the memory location whose key is x, is val with respect to the consistency level $level$.

.

We denote the set of all variables in the read-write store by $Vars$ and the set of all values forming the contents of the read-write store by the set of natural numbers $\mathbb{N}$. We assume that the initial value of all the variables is 0. Let $\mathbb{N}^+$ denote the set $\mathbb{N} \setminus \{0\}$.

For simplicity, we assume only two consistency levels $\{\mathsf{strong}, \mathsf{weak}\}$.

If o is an instance of an operation performed by the application on the read-write store, then

- $Op(o) \in \{\mathsf{Read}, \mathsf{Write}\}$ indicates whether the operation o is a Read or a Write operation.
- $Var(o) \in Vars$ denotes the variable on which the Read/Write operation is being performed.
- $Args(o) \in \mathbb{N}^+ \cup \{\texttt{Nil}\}$ denotes the arguments passed to the operation invocation. $Args(o) = \texttt{Nil}$ iff $Op(o) = \mathsf{Read}$ and $Args(o) \in \mathbb{N}^+$ iff $Op(o) = \mathsf{Write}$.
- $Ret(o) \in \mathbb{N} \cup \{\texttt{Nil}\}$ denotes the return value obtained as a part of the operation response. $Ret(o) = \texttt{Nil}$ iff $Op(o) = \mathsf{Write}$ and $Ret(o) \in \mathbb{N}$ iff $Op(o) = \mathsf{Read}$.
- If $Op(o) = \mathsf{Read}$ then, $Level(o) \in \{\mathsf{strong}, \mathsf{weak}\}$ denotes the level of the read operation.

The behaviour of the read-write data-store as observed by the application is the sequence of reads and writes that it performs on the stores. This related sequence of read and write operation performed by the application is termed a *session*. Thus the behaviour of the read-write store seen by each session is a total order of read/write operations performed in that session.

The behaviour of the read-write store is the collection of behaviours seen by all the sessions. In Figure 1 we can see the behaviour of the data-store as observed by the three sessions accessing the data-store. We will call this behaviour a *hybrid history*, formally defined as follows:

Definition 1 (Hybrid History) *A hybrid history of a read-write store is the tuple* $H = (\mathcal{O}, \mathsf{so})$ *where* $\mathcal{O}$ *is the set of read-write operations and* so *is a collection of total orders where each total order is a session-order.*

For a history H*, we define the following subsets of* $\mathcal{O}$*.*

- $\mathcal{O}_{\mathsf{Read}} = \{o \in \mathcal{O} \mid Op(o) = \mathsf{Read}\}$ *is the set of read operations.*
- $\mathcal{O}_{\mathsf{Write}} = \{o \in \mathcal{O} \mid Op(o) = \mathsf{Write}\}$ *is the set of write operations.*
- $\mathcal{O}_{\mathsf{weak}} = \mathcal{O}_{\mathsf{Write}} \cup \{o \in \mathcal{O} \mid Level(o) = \mathsf{weak}\}$ *is the set of weak operations.*
- $\mathcal{O}_{\mathsf{strong}} = \mathcal{O}_{\mathsf{Write}} \cup \{o \in \mathcal{O} \mid Level(o) = \mathsf{strong}\}$ *is the set of strong operations.*

The weak fragment of the history H *is denoted* H_{weak} *and defined to be* $(\mathcal{O}_{\mathsf{weak}}, \mathsf{so}|_{\mathcal{O}_{\mathsf{weak}}})$*. Similarly the strong fragment of the history* H *is denoted* H_{strong} *and is defined to be* $(\mathcal{O}_{\mathsf{strong}}, \mathsf{so}|_{\mathcal{O}_{\mathsf{strong}}})$*. When we say a* well-defined fragment *of the history* H*, we refer to either* H*,*H_{weak} *or* H_{strong}*.*

Note that we take the write operations to be part of both the strong and weak fragments.

If $X, Y \subseteq \mathcal{O} \times \mathcal{O}$ then, we use the following notations to define some well-defined binary relations over $\mathcal{O}$ involving X, Y.

- For $\mathsf{op} \in \{\mathsf{Read}, \mathsf{Write}\}$, $(X)_{\mathsf{op}} = X \cap (\mathcal{O}_{\mathsf{op}} \times \mathcal{O}_{\mathsf{op}})$
- For $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$, $(X)_{\ell} = X \cap (\mathcal{O}_{\ell} \times \mathcal{O}_{\ell})$
- $X; Y$ denotes the relation obtained by composition of X and Y defined as $X; Y = \{(x, y) \mid \exists z : (x, z) \in X \wedge (z, y) \in Y\}$.
- Finally $\mathsf{total}(X)$ indicates that the relation X is a total order.

Now, when a replica of the read-write store receives operations from the applications it decides how the effects of the older operations known to the replica, either by the virtue of having received them from applications, or from other replicas of the data-store, should be made visible to the new operation. This is abstracted by a visibility relation over the history, which defines for every operation in the history, which other operations of the history are visible to it.

Definition 2 (Visibility Relation) *A visibility relation* vis *over a history* $H = (\mathcal{O}, \mathsf{so})$ *is an acyclic relation over* $\mathcal{O}$*. For* $o, o' \in \mathcal{O}$*, we write* $o \xrightarrow{\mathsf{vis}} o'$ *to indicate that the effects of the operation* o *are visible to the operation* o'*.*

If a pair of operations o, o' *are not related by* vis*, we term them concurrent operations, denoted by* $o||_{\mathsf{vis}}o'$*.*

We define the View *of an operation* o*, denoted by* $\mathsf{View}_{\mathsf{vis}}(o)$ *to be the set of all the* Write *operations visible to it.*

For the history in Figure 1, we can define a visibility relation to be

$$\{A \xrightarrow{\mathsf{vis}} B, G \xrightarrow{\mathsf{vis}} C, G \xrightarrow{\mathsf{vis}} D, H \xrightarrow{\mathsf{vis}} D, G \xrightarrow{\mathsf{vis}} E, H \xrightarrow{\mathsf{vis}} E, I \xrightarrow{\mathsf{vis}} E, G \xrightarrow{\mathsf{vis}} F\}$$

When the replicas communicate with each other, they need to reconcile the effects of concurrent write operations in order to converge to the same state eventually. In case of convergent data-stores this is done using some technique such

as *Last Writer Wins* which totally orders all write operations. This is abstracted by an arbitration relation, which is a total order over all write operations in the history. We will denote by arb the arbitration relation. We assume that the arbitration relation is consistent with the visibility relation, in the sense that for a pair of writes o and o', if o is visible to o' then o is arbitrated before o'.

Definition 3 (Arbitration Relation) *An arbitration relation* arb *over a hybrid history* $H = (\mathcal{O}, \mathsf{so})$ *is a total order over* $\mathcal{O}_{\mathsf{Write}}$. *For* $o_i, o_j \in \mathcal{O}$, *we say* $o_i \xrightarrow{\mathsf{arb}} o_j$ *to indicate that operation* o_i *has been arbitrated before the operation operation* o_j.

For the history in Figure 1 the arbitration relation can be the total order defined by:

$$A \xrightarrow{\mathsf{arb}} G \xrightarrow{\mathsf{arb}} H \xrightarrow{\mathsf{arb}} I$$

The consistency criteria enforce some constraints over the choice of the visibility and arbitration relations. These constraints are defined in terms of axioms. We shall define these axioms using a grammar adapted from [8].

Definition 4 (Grammar for Consistency Criteria) *Consistency criteria are given by the set* Φ_c *generated by the following grammar:*

- $\tau \in \mathit{RelTerms} := \mathtt{so} \mid \mathtt{vis} \mid \tau;\tau$
- $\beta \in \Phi_{vis} := \top \mid \mathsf{total}(\mathtt{vis}) \mid \tau \subseteq \mathtt{vis} \mid \beta \wedge \beta$
- $\gamma \in \Phi_{arb} := \top \mid (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$
- $\alpha \in \Phi_c := \beta \wedge \gamma$

For $\alpha = \beta \wedge \gamma \in \Phi_c$, we define $\mathit{VisForm}(\alpha)$ and $\mathit{ArbForm}(\alpha)$ to be β and γ, respectively. $\mathit{RelTerms}(\alpha) = \{\tau \in \mathit{RelTerms} \mid \tau \subseteq \mathtt{vis}$ is a subformula of $\alpha\}$.

We define $\mathit{VisBasic}(\alpha)$ to be the maximal fragment of α containing only subformulas of the type $\tau \subseteq \mathtt{vis}$. Thus,

$$\mathit{VisBasic}(\alpha) = \bigwedge_{\tau \in \mathit{RelTerms}(\alpha)} \tau \subseteq \mathtt{vis}$$

In general, each of $\mathtt{so}$, $\mathtt{vis}$ and $\mathtt{arb}$ are variables to be substituted by some binary relation over the set of operations in the history.

Suppose $H = (\mathcal{O}, \mathsf{so})$ is a history, $X, Y, Z \subseteq \mathcal{O} \times \mathcal{O}$ are binary relations over the set of operations, and $\alpha \in \mathtt{axioms}$. We say that $X, Y, Z \models \alpha$ iff $\alpha[\mathtt{so} := X, \mathtt{vis} := Y, \mathtt{arb} := Z]$ is true.

We now define a consistency criterion in terms of the grammar.

Definition 5 (Consistency Criterion in a history) *Suppose* $H_\ell = (\mathcal{O}_\ell, \mathsf{so}_\ell)$ *is a well defined fragment of a hybrid history, and* vis_ℓ *and* arb *are respectively the visibility and arbitration relations defined over* H. *A consistency criteria is a formula* $\alpha \in \Phi_c$. *We say that* $H_\ell, \mathsf{vis}_\ell, \mathsf{arb} \models \alpha$ *when* $\mathsf{so}_\ell, \mathsf{vis}_\ell, \mathsf{arb} \models \alpha$.

Some well known consistency criteria are given below:

– **Basic Eventual Consistency(BEC)**

$$\mathsf{BEC} := \top$$

– **Read Your Writes (RYW)**

$$\mathsf{RYW} := \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$$

– **Monotonic Reads (MR)**

$$\mathsf{MR} := \mathtt{vis}; \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$$

– **Monotonic Read Writes (MW)**

$$\mathsf{MW} := \mathtt{so}; \mathtt{vis} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$$

– **Strong Eventual Consistency (SEC)**

$$\mathsf{SEC} := \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ \mathtt{vis}; \mathtt{so} \subseteq \mathtt{vis}$$

– **FIFO Consistency (FIFO)**

$$\mathsf{FIFO} := \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ \mathtt{vis}; \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ \mathtt{so}; \mathtt{vis} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$$

– **Causal Consistency (CC)**

$$\mathsf{CC} := \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ \mathtt{vis}; \mathtt{vis} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$$

– **Sequential Consistency (SEQ)**

$$\mathsf{SEQ} := \mathtt{so} \subseteq \mathtt{vis} \ \wedge \ \mathtt{vis}; \mathtt{vis} \subseteq \mathtt{vis} \ \wedge \ (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb} \ \wedge \ \mathsf{total}(\mathtt{vis})$$

We say that a consistency criteria α is at least as *strong* as another consistency criteria α' if for every history H, visibility relation vis, and arbitration relation arb over H, if $H, \mathsf{vis}, \mathsf{arb} \models \alpha$ then $H, \mathsf{vis}, \mathsf{arb} \models \alpha'$.

Suppose $H = (\mathcal{O}, \mathsf{so})$ is a history with fragments H_{weak} and H_{strong}. Let α_w and α_s respectively be the weak and strong consistency criteria. Then we want to choose weak and strong visibility relations vis_w, vis_s respectively and arbitration relations arb such that $H_{\mathsf{weak}}, \mathsf{vis}_w, \mathsf{arb} \models \alpha_w$ and $H_{\mathsf{strong}}, \mathsf{vis}_s, \mathsf{arb} \models \alpha_s$.

As we had noted in the previous section, in a multilevel setting, it is not sufficient to separately satisfy the axioms corresponding to the weak and strong consistency criteria. We now define the multilevel visibility constraints as a conjunction of formulas.

Definition 6 (Multilevel Constaints) *The set of multilevel constraints $\Phi_{multilevel}$ is a finite conjunction $\phi = \psi_1 \wedge \cdots \wedge \psi_n$, where each ψ_i is one of the following formulas:*

– $\psi^{ext}_{strong} := (\mathtt{vis}^{\mathtt{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathtt{vis}^{\mathtt{strong}}$
– $\psi^{ext}_{weak} := (\mathtt{vis}^{\mathtt{strong}}; \mathsf{so})_{\mathsf{weak}} \subseteq \mathtt{vis}^{\mathtt{weak}}$

- $\psi^{rest}_{strong} := (\mathtt{vis}^{\mathtt{strong}}) \subseteq (\mathtt{vis}^{\mathtt{weak}}; \mathtt{so})_{\mathsf{strong}}$
- $\psi^{rest}_{weak} := \mathtt{vis}^{\mathtt{weak}} \subseteq (\mathtt{vis}^{\mathtt{strong}}; \mathtt{so})_{\mathsf{weak}}$
- $\psi^{mr}_{strong} := (\mathtt{vis}^{\mathtt{strong}}; \mathtt{so})_{\mathsf{strong}} \subseteq \mathtt{vis}^{\mathtt{strong}}$
- $\psi^{mr}_{weak} := (\mathtt{vis}^{\mathtt{weak}}; \mathtt{so})_{\mathsf{weak}} \subseteq \mathtt{vis}^{\mathtt{weak}}$

Suppose $H = (\mathcal{O}, \mathsf{so})$ is a history and $X, Y, Y' \subseteq \mathcal{O} \times \mathcal{O}$ are binary relations over the set of operations and ϕ is a multilevel visibility constraint. We say that $X, Y, Y' \models \phi$ iff $\phi[\mathtt{so} := X, \mathtt{vis}^{\mathtt{weak}} := Y, \mathtt{vis}^{\mathtt{strong}} := Y']$ is true. if $\phi = \psi_1 \wedge \psi_2 \cdots \wedge \psi_n$ we say that each ψ_i is a subformula of ϕ.

The formula ψ^{ext}_{strong} denotes *strong-extension* and it insists that the strong operations see the effects seen by the prior weak operations in the session. Similarly, the formula ψ^{ext}_{weak} denoting *weak-extension* insists that the weak operations see the effects seen by the prior strong operations in the session. These two guarantee that the effect seen by reads of one consistency level remain monotonically visible to the subsequent reads of another consistency level. The formula ψ^{rest}_{strong} denoting *strong-restriction* insists that the effects visible to any strong operation is only the subset of the effects visible to the prior weak operations. Similarly, ψ^{rest}_{weak} denoting *weak-restriction* makes sure that the effects visible to any weak operation is only the subset of the effects visible to the prior strong operations. Finally the formulas ψ^{mr}_{strong} (resp. ψ^{mr}_{weak}) denoting *strong-monotonic-reads* (resp. *weak-monotonic-reads*) make sure that the effects visible to the prior strong (resp. weak) operations remain visible to strong (resp. weak) operations later on in that session.

We say that the multilevel constraint ϕ is well-defined if:

- If ψ^{rest}_{strong} is a subformula of ϕ then ψ^{mr}_{weak} is also a subformula of ϕ.
- If ψ^{rest}_{weak} is a subformula of ϕ then ψ^{mr}_{strong} is also a subformula of ϕ.

We say that a hybrid history $H = (\mathcal{O}, \mathsf{so})$ along with weak and strong visibility relations vis_w and vis_s satisfies multilevel constraint ϕ, written as $H, \mathsf{vis}_w, \mathsf{vis}_s \models \phi$ iff $\mathsf{so}, \mathsf{vis}_w, \mathsf{vis}_s \models \phi$.

Now Cassandra's multilevel consistency from the example in the prior section required that the weaker reads should see effects seen by the prior strong reads. This can be modelled by the formula ψ^{ext}_{weak}.

The multilevel consistency expected by the Incremental Consistency Guarantees (ICG) library from [12] requires that the weak reads see the effects of prior strong reads and the strong reads see the effects seen by prior weak reads. This can be modelled as the formula $\psi^{ext}_{weak} \wedge \psi^{ext}_{strong}$. Finally the Bolt-On consistency from [1] requires that the strong reads only see a subset of the effects seen by the prior weak reads. This can be modelled as $\psi^{rest}_{strong} \wedge \psi^{mr}_{weak}$.

In order to explain the correctness of a Hybrid history, we need to define correctness in terms of the specification of read-write stores.

Let H be a well defined fragment of a hybrid history. Let vis and arb be visibility and arbitration relations over H.

We say that a write operation o' is a *related-write* of a read operation o iff o' is in the view of o and both o and o' operate on the same variable. The

set of all related writes of o, denoted as $RelWrites_{\mathsf{vis}}(o)$ is defined to be $\{o' \in \mathsf{View}_{\mathsf{vis}}(o) \mid Var(o) = Var(o')\}$.

The maximal among these related writes with respect to the visibility relation vis is denoted by $MaxRelWrites_{\mathsf{vis}}(o)$, defined to be the set

$$\{o' \in RelWrites_{\mathsf{vis}}(o) \mid \forall o'' \in RelWrites_{\mathsf{vis}}(o) : o'' \xrightarrow{\mathsf{vis}} o' \vee o'' ||_{\mathsf{vis}} o'\}$$

The effective write of a read-operation o, denoted by $EffWrite^{\mathsf{arb}}_{\mathsf{vis}}(o)$ is defined to be the maximum write operation from the set of maximal related writes of o arbitrated as per the arbitration relation.

$$EffWrite^{\mathsf{arb}}_{\mathsf{vis}}(o) = \begin{cases} max(\mathsf{arb}|MaxRelWrites_{\mathsf{vis}}(o)) & \text{if } MaxRelWrites_{\mathsf{vis}}(o) \neq \emptyset \\ \bot & \text{otherwise} \end{cases}$$

Definition 7 (Correctness with respect to Read-Write Specification) *A well defined fragment $H = (\mathcal{O}, \mathsf{so})$ of a hybrid history with visibility relations* vis *and* arb *defined over it is said to be correct with respect to the read-write specification iff for every read operation o in $\mathcal{O}$*

- $EffWrite^{\mathsf{arb}}_{\mathsf{vis}}(o) = \bot$ *iff* $Ret(o) = 0$
- *If* $o' = EffWrite^{\mathsf{arb}}_{\mathsf{vis}}(o)$ *then* $Ret(o) = Args(o')$.

We write $H, \mathsf{vis}, \mathsf{arb} \models \mathsf{Spec}_{\mathsf{RW}}$ to indicate that the fragment H along with visibility relation vis *and arbitration relation* arb *is correct with respect to the Read-Write Specification.*

We now formally define when a Hybrid History is deemed to be correct.

Definition 8 (Multilevel Correctness of Hybrid History) *A Hybrid History $H = (\mathcal{O}, \mathsf{so})$ of a Read-Write store with is said to be* multilevel correct *with respect to a* weak *consistency criterion α_w,* strong *consistency criterion α_s and multilevel consistency constraint ϕ, iff there exists visibility relations* vis_w *and* vis_s *over H_{weak} and H_{strong} respectively and arbitration relation* arb *such that*

- $H_{\mathsf{weak}}, \mathsf{vis}_w, \mathsf{arb} \models \alpha_w, \mathsf{Spec}_{\mathsf{RW}}$
- $H_{\mathsf{strong}}, \mathsf{vis}_s, \mathsf{arb} \models \alpha_s, \mathsf{Spec}_{\mathsf{RW}}$
- $H, \mathsf{vis}_w, \mathsf{vis}_s \models \phi$.

4 Testing Multilevel Correctness of a Hybrid History

Given a read-write hybrid history $H = (\mathcal{O}, \mathsf{so})$ whose multi-level correctness we want to test with respect to weak and strong consistency criteria $\alpha_w = \beta_w \wedge \gamma_w$ and $\alpha_s = \beta_s \wedge \gamma_s$ and multilevel constraints given by ϕ.

We note that for the history to be correct for every non-initial read operation there should exist a write operation writing the exact same value to the variable read by the read operation. Suppose we term this relation as the *reads-from* relation associating a write operation to the read who reads its effect.

Our strategy for testing the multilevel correctness of H would be to enumerate all such reads-from relation rf, for each of which we find visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ respectively containing $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ such that they satisfy the visibility constraints imposed by the individual consistency criteria as well as the multilevel constraints, i.e $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$ and $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$. We then check for the presence of a finite number of *bad-patterns* in these visibility relations. If any of the bad-patterns exist, the it implies that for every arbitration relation arb, either the arbitration constraints γ_w or γ_s is not satisfied, or one of the fragments fails to satisfy the correctness with respect to the read-write specification $\mathsf{Spec}_{\mathsf{RW}}$.

We repeat this for all possible reads-from relations for the history. If the History is multi-level correct, then we will find a witness reads-from relation rf and visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ extending it such that all the constraints are satisfied and has no bad-patterns. Otherwise, every pair of weak and strong visibility relation extending every reads-from relation has some bad-pattern.

We will first present the bad-pattern characterization for multilevel correctness of a hybrid history in the next subsection. In the subsection 4.2 we provide a procedure for computing the minimal visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ for a given reads-from relation rf that satisfies β_w , β_s and ϕ.

4.1 Bad Pattern characterization for multilevel correctness

We now present a characterization for the correctness of Hybrid Histories based on the non-existence of certain bad patterns. This is a generalization of the Bad-Pattern characterization presented for causal consistency in [3].

Given a hybrid history, we can associate each Read with a unique write operation from the history whose effect the Read operation reads from. We call this the *Reads-From* relation.

Definition 9 (Reads-From) *A reads-from relation* rf *over a history* $H = (\mathcal{O}, \mathsf{so})$ *is a binary relation such that*

1. $(o_i, o_j) \in \mathsf{rf} \implies Op(o_i) = \mathsf{Write}, Op(o_j) = \mathsf{Read}, Var(o_i) = Var(o_j), Args(o_i) = Ret(o_j)$
2. $(o_i, o_j) \in \mathsf{rf} \wedge (o_k, o_j) \in \mathsf{rf} \implies o_i = o_k$.
3. $\forall o_i : (o_i, o_j) \notin \mathsf{rf} \implies \forall o_k : Op(o_k) = \mathsf{Read} \vee Var(o_k) \neq Var(o_j) \vee Args(o_k) \neq Ret(o_j)$

Condition 1 associates a read operation with a write operation only if they operate on the same variable and that the return value of the read operation matches the argument of the write operation.

Condition 2 ensures that a read operation is associated with at most one write operation.

Finally Condition 3 insists that if a read-operation doesn't have a matching write operation, it is only because there is no such matching write operation in the hybrid history.

Let rf be a reads-from relation on a Hybrid History $H = (\mathcal{O}, \mathsf{so})$. For a Read operation $o \in \mathcal{O}$, if there exists a Write operation o' such that $(o', o) \in \mathsf{rf}$, then we say that $\mathsf{rf}^{-1}(o) = o'$. Suppose no such o' exists, then we set $\mathsf{rf}^{-1}(o) = \bot$.

Further, we denote by $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ the reads-from relation restricted to H_{weak} and H_{strong} respectively.

Suppose rf_ℓ is a reads-from relation over the well-defined fragment H_ℓ. We say that a visiblity relation vis_ℓ over H_ℓ *extends* rf_ℓ iff $\mathsf{rf}_\ell \subseteq \mathsf{vis}_\ell$. Suppose arb is an arbitration relation over H_ℓ. Then, we say that $(\mathsf{vis}_\ell, \mathsf{arb})$ *realize* rf_ℓ iff for all read operations $o \in \mathcal{O}_\ell$, $\mathsf{rf}^{-1}(o) = \mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o)$.

Given a reads-from relation and a visibility relation that extends it, we can define a conflict relation that orders all the remaining maximal related writes of a read behind the write that the read *reads-from*. The conflict relation captures the essence of the arbitration relation for a given reads-from relation and a visibility relation extending it.

Definition 10 (Conflict Relation) *Let $H_\ell = (\mathcal{O}_\ell, \mathsf{so}_\ell)$ be a well-defined fragment of a hybrid history. Let rf_ℓ be a reads-from relation over H_ℓ. Let vis_ℓ be a visiblity relation over H_ℓ that extends rf_ℓ. We define the conflict relation for rf_ℓ and vis_ℓ, denoted $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell)$, as the set*

$$\{(o'', o') \mid \exists o \in \mathcal{O}_\ell : \mathit{Op}(o) = \mathsf{Read} \wedge o'', o' \in \mathit{MaxRelWrites}_{\mathsf{vis}}(o) \wedge o' = \mathsf{rf}^{-1}(o)\}.$$

We shall define the bad patterns that characterize the correctness of the hybrid history.

Definition 11 (Bad Patterns for a hybrid history) *Let $H = (\mathcal{O}, \mathsf{so})$ be a hybrid history with weak and strong consistency criteria $\alpha_w = \beta_w \wedge \gamma_w$ and $\alpha_s = \beta_s \wedge \gamma_s$ respectively and multilevel constraints ϕ. Let rf be a reads-from relation over H. Let $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ be the weak and strong visibility relations extending $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ respectively, such that $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$ and $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$. We define the following bad patterns*

- $\mathsf{BADVISIBILITY} : \bigvee_{\ell \in \{\mathsf{weak}, \mathsf{strong}\}} \mathsf{Cyclic}(\mathsf{vis}_\ell)$
- $\mathsf{THINAIR} : \exists o \in \mathcal{O} : \mathit{Op}(o) = \mathsf{Read} \wedge \mathit{Ret}(o) \neq 0 \wedge \mathsf{rf}^{-1}(o) = \bot$
- $\mathsf{BADINITREAD}$*:*

$$\bigvee_{\ell \in \{\mathsf{weak}, \mathsf{strong}\}} \exists o \in \mathcal{O} : \mathit{Op}(o) = \mathsf{Read} \wedge \mathit{Level}(o) = \ell \wedge \mathit{Ret}(o) = 0 \wedge \mathit{RelWrites}_{\mathsf{vis}_\ell}(o) \neq \emptyset$$

- $\mathsf{BADREAD}$*:*

$$\bigvee_{\ell \in \{\mathsf{weak}, \mathsf{strong}\}} \exists o \in \mathcal{O} : \mathit{Op}(o) = \mathsf{Read} \wedge \mathit{Level}(o) = \ell \wedge \mathsf{rf}^{-1}(o) \notin \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$$

- BADARB*:*

$$\mathsf{Cyclic}\Big(\bigcup_{\ell \in \{\mathsf{weak}, \mathsf{strong}\}} (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \cup (\mathsf{vis}_\ell)_{\mathsf{Write}})\Big)$$

BADVISIBILITY says that one of the visibility relations has a cycle.

THINAIR says that there exists a read in the history which reads a non-initial value which is not written to by any write operation in the hybrid history.

BADINITREAD says that there is a read operation on a variable which reads the initial value despite having non-initial write to that variable in its view.

BADREAD says that the write operation from which the read-operation reads is not a maximal write, and there are other writes in the view of the read operation that would have overwritten the value written by that write.

BADARB says that the union of the conflict relations along visibility relation restricted to only the Write operations has a cycle hinting that the arbitration relation might have a cycle, and hence is not a total order.

We will now provide a result characterizing multi-level correctness of a hybrid history in terms of non-existence of these bad patterns. We prove this in Appendix A.

Theorem 12 (bad patterns characterization). *A hybrid history $H = (\mathcal{O}, \mathsf{so})$ is said to be multilevel correct with respect to weak and strong consistency criteria $\alpha_w = \beta_w \wedge \gamma_w$ and $\alpha_s = \beta_s \wedge \gamma_s$ and multilevel constraint ϕ iff there exists a reads-from relation rf and visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ that extend $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ respectively such that $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$ and $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$ and no bad pattern exists in H.*

4.2 Constructing Minimal Visibility Relations

Suppose $H = (\mathcal{O}, \mathsf{so})$ is a hybrid history. Let $\alpha_w = \beta_w \wedge \gamma_w$, $\alpha_s = \beta_s \wedge \gamma_s$ be the formulas defining the weak and strong consistency criteria, and let ϕ be the formula defining the multilevel constraints. Let $\beta'_w = \mathit{VisBasic}(\beta_w)$ and $\beta'_s = \mathit{VisBasic}(\beta_s)$

We will now provide a procedure for constructing a minimal visibility relation extending a reads-from relation that satisfies the visibility constraints from β_w, β_s and ϕ. The pseudo-code for the procedure is presented in Algorithm 1 and 2.

In Algorithm 1, we use the notation $o' \xrightarrow[\ell]{\mathsf{so}(1)} o$ to mean that o' is the nearest operation with level ℓ preceding o in its session. Thus $o' \xrightarrow[\ell]{\mathsf{so}(1)} o$ iff the following conditions hold:

- $o' \xrightarrow{\mathsf{so}} o$,
- $\mathit{Level}(o') = \ell$, and
- $\forall o'' : o'' \xrightarrow{\mathsf{so}} o \wedge \mathit{Level}(o'') = \ell \implies o'' = o' \vee o'' \xrightarrow{\mathsf{so}} o' \xrightarrow{\mathsf{so}} o$.

In Lines 1-12 we have a method BuildMinVisSingle that takes as input a visiblity relation vis_ℓ for a well defined fragment of history $(\mathcal{O}_\ell, \mathsf{so}_\ell)$ and constructs an extension $\mathsf{vis}_{\mathsf{new}}$ that satisfies the formula $\mathit{VisBasic}(\alpha_\ell)$. We achieve this by iterating over the *RelTerms* appearing in $\mathit{RelTerms}(\alpha_\ell)$ (Line 6) and extending the previous visibility relation $\mathsf{vis}_{\mathsf{prev}}$ with the evaluation of the term (Line 7). We do this until we obtain a relation $\mathsf{vis}_{\mathsf{new}}$ which we can no longer extend

Algorithm 1 Constructing minimal visibility relations

```
1  BuildMinVisSingle(O_ℓ, so_ℓ, vis_ℓ, α_ℓ):
2      Let vis_old := vis_ℓ;
3
4      while (True):
5          Let vis_prev := vis_old;
6          for τ ∈ RelTerms(α_ℓ)):
7              vis_new := vis_prev ∪ τ[so_ℓ, vis_prev];
8              vis_prev := vis_new;
9          if (vis_new == vis_old)
10             return vis_new
11         vis_old := vis_new
12
13
14 BuildMinVisMulti(O, so, vis_weak, vis_strong, ψ)
15     if ψ ∈ {ψ^ext_strong, ψ^rest_strong, ψ^mr_strong}:
16         Let ℓ = strong, ℓ' = weak;
17     else if ψ ∈ {ψ^ext_weak, ψ^rest_weak, ψ^mr_weak}:
18         Let ℓ = weak, ℓ' = strong;
19
20     Let vis^old_ℓ := vis_ℓ;
21     Let vis^old_ℓ' := vis_ℓ';
22
23     if ψ ∈ {ψ^ext_strong, ψ^ext_weak}:
24         Let vis^new_ℓ := vis^old_ℓ ∪ (vis^old_ℓ'; so)_ℓ;
25         Let vis^new_ℓ' := vis^old_ℓ';
26     else if ψ ∈ {ψ^mr_strong, ψ^mr_weak}:
27         Let vis^new_ℓ := vis^old_ℓ ∪ (vis^old_ℓ; so)_ℓ;
28         Let vis^new_ℓ' := vis^old_ℓ';
29     else if ψ ∈ {ψ^rest_strong, ψ^rest_weak}:
30         Let UnaccountedWrites := vis^old_ℓ \ vis^old_ℓ'; so;
31         Let NearestPriorOther := {(o, o') | o ∈ O_ℓ and o' --so(1)/ℓ'--> (o)}
32         vis^new_ℓ' := vis^old_ℓ' ∪ UnaccountedWrites; NearestPriorOther
33         vis^new_ℓ := vis^old_ℓ
34
35      if ψ ∈ {ψ^ext_weak, ψ^rest_weak, ψ^mr_weak}:
36         return (vis^new_ℓ, vis^new_ℓ')
37     else if ψ ∈ {ψ^ext_strong, ψ^rest_strong, ψ^mr_weak}:
38         return (vis^new_ℓ', vis^new_ℓ)
39
40
41 ComputeVisSet(O_ℓ, so_ℓ, vis_ℓ, α_ℓ)
42     if total(vis) is a subformula in α_ℓ:
43         visSet_ℓ := {totvis|totvis is a total order over O_ℓ such that vis_ℓ ⊆ totvis}
44     else :
45         visSet_ℓ := {vis_ℓ}
46
47     return visSet_ℓ
48
```

(Line 9). This final visibility relation $\mathsf{vis}_{\mathsf{new}}$ extends vis_ℓ and satisfies the formula $VisBasic(\alpha_\ell)$.

In Lines 14-39, we have the procedure BuildMinVisMulti which takes as inputs the hybrid history $(\mathcal{O}, \mathsf{so})$, visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ and an individual conjunct ψ appearing in ϕ. Note that the multilevel constraints relates the write operations visible to the operations of level ℓ in terms of the writes seen by operations of level ℓ' that have occured previously in the session. Depending on the conjunct ψ, we set ℓ and ℓ' appropriately(Lines 15-18). If ψ is either ψ^{ext}_{strong} or ψ^{ext}_{weak} then, we extend the visibility relation for level ℓ by relating each ℓ-operation to the Writes that have been seen by any of the ℓ'-operations prior to the ℓ-operation in its session (Line 24). The visibility relation for level ℓ' remains unchanged in this case (Line 25). If ψ is either ψ^{mr}_{strong} or ψ^{mr}_{weak}, we just ensure that the visibility relation at level ℓ satisfies monotonic reads (Lines 26-28). On the other hand, if ψ is either ψ^{rest}_{strong} or ψ^{rest}_{weak}, then, we need to ensure that any Writes visible to a ℓ- operation is visible to some previous ℓ'-operation in its session. Now, since in a well-defined ϕ the presence of ψ^{rest}_{strong} (resp. ψ^{rest}_{weak}) also implies the presence of the conjunct ψ^{mr}_{weak} (resp. ψ^{mr}_{strong}), it means that any write visible to any ℓ'-operation is visible to all the subsequent ℓ'-operations in its session. Thus, any Write visible to a ℓ-operation should be visible to its nearest preceding ℓ'-operation in its session. We first compute the unaccounted Writes seen by ℓ-operations that haven't been seen by any of the prior ℓ'- operations (Line 30). We then compute the nearest ℓ'- predecessor for each ℓ-operation (Line 31). We then extend the visibility relation for ℓ' by relating these unaccounted writes seen by each of of each ℓ-operation to the nearest preceding ℓ'-operation in its session(Line 32). In this case the visibility relation for ℓ remains unchanged (Line 33). We return these extended visibility relations as a pair, where the weak visibility extension is followed by strong visiblity extension (Lines 35-38).

In Lines 49-67 we have the procedure ComputeStableExtension which takes history $(\mathcal{O}, \mathsf{so})$ a pair of visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ and extends it to $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ such that they individually satisfy $VisBasic(\alpha_w)$ (Line 55) and $VisBasic(\alpha_s)$ (Line 57) respectively and jointly satisfy ϕ (Lines 59-62). We repeat this till we can extend these relations no longer, which implies that they have satisfied all the constraints (Lines 64-65).

The procedure TestMultiLevelCorrectness in Lines 79-96 takes as input a hybrid history $H = (\mathcal{O}, \mathsf{so})$ whose multilevel correctness we want to check with respect to formulas α_w , α_s and ϕ.

We first check if the History has a bad-pattern for multilevel constraint ψ^{rest}_{strong} (resp. ψ^{rest}_{weak}) where we have a strong-Read (resp. strong-Read) which returns a non-initial value, but has no prior weak (resp. strong) operations in its session (Lines 80-81, 69-77).

We first enumerate the set of possible reads-from relations on the history (line 82). We then iterate through each of the Reads-from relations rf to see whether it can be extended to construct a minimal visibility relation satisfying all the constraints and having no bad-patterns (Lines 83-94). For each rf, we con-

Algorithm 2 Testing multilevel correctness of a hybrid history

```
49  ComputeStableExtension(O, so, vis_weak, vis_strong, α_w, α_s, φ):
50      Let vis^old_weak := vis_weak, vis^old_strong := vis_strong
51
52      while (True):
53          Let vis^prev_weak := vis^old_weak, vis^prev_strong := vis^old_strong
54
55          Let vis^new_weak := BuildMinVisSingle(O_weak, so_weak, vis^prev_weak, α_w);
56
57          Let vis^new_strong := BuildMinVisSingle(O_strong, so_strong, vis^prev_strong, α_s);
58
59          for each subformula ψ_i in the conjunction φ:
60              vis^prev_weak := vis^new_weak, vis^prev_strong := vis^new_strong
61
62              (vis^new_weak, vis^new_strong) = BuildMinVisMulti(O, so, vis^prev_weak, vis^prev_strong, ψ_i)
63
64          if vis^new_weak = vis^old_weak and vis^new_strong = vis^old_strong:
65              return (vis^new_weak, vis^new_strong)
66
67          vis^old_weak := vis^new_weak, vis^old_strong := vis^new_strong
68
69  CheckBadRestrictionHistory(O, so, φ):
70      if ψ^rest_strong is a formula in the conjunction φ:
71          if ∃o ∈ O_strong : Op(o) = Read and Ret(o) ≠ 0 and --so(1)-->_weak (o) = ⊥:
72              return BadPatterns
73      if ψ^rest_weak is a formula in the conjunction φ:
74          if ∃o ∈ O_weak : Op(o) = Read and Ret(o) ≠ 0 and --so(1)-->_strong (o) = ⊥:
75              return BadPatterns
76
77      return NoBadPatterns
78
79  TestMultiLevelCorrectness(O, so, α_w, α_s, φ):
80      if CheckBadRestrictionHistory(O, so, φ) = BadPatterns:
81          return BadHistory
82      Let rfSet := {rf|rf is a reads-from relation over (O, so)}
83      for rf ∈ rfSet:
84          Let vis^min_weak := BuildMinVisSingle(O_weak, so_weak, rf_weak, α_w);
85          Let visSet_weak = ComputeVisSet(O_weak, so_weak, vis^min_weak, α_w);
86
87          Let vis^min_strong := BuildMinVisSingle(O_weak, so_strong, rf_strong, α_s);
88          Let visSet_strong = ComputeVisSet(O_strong, so_strong, vis^min_strong, α_s);
89
90          for vis_weak ∈ visSet_weak, vis_strong ∈ visSet_strong:
91              Let (vis^stable_weak, vis^stable_strong) := ComputeStableExtension(O, so, vis_weak, vis_strong,
                                                                                   α_w, α_s, φ);
92
93              if CheckBadPatterns(O, so, rf, vis^stable_weak, vis^stable_strong) = NoBadPatterns:
94                  return (rf, vis^stable_weak, vis^stable_strong)
95
96      return BadHistory
```

struct minimal visibility relations $\mathsf{vis}^{\min}_{\mathsf{weak}}$ and $\mathsf{vis}^{\min}_{\mathsf{strong}'}$ extending $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}'}$ respectively and satisfying the subformulas $VisBasic(\alpha_w)$ and $VisBasic(\alpha_s)$ respectively (Lines 84,87).

If α_w (resp. α_s) contains the subformula $\mathsf{total}(\mathtt{vis})$, we enumerate the set of all the total orders extending $\mathsf{vis}^{\min}_{\mathsf{weak}}$ (resp. $\mathsf{vis}^{\min}_{\mathsf{strong}}$) in the set $\mathsf{visSet}_{\mathsf{weak}}$ (resp. $\mathsf{visSet}_{\mathsf{strong}}$) in Line 85 (resp. Line 88). If α_w (resp. C_{strong}) does not contain the subformula $\mathsf{total}(\mathtt{vis})$, then, $\mathsf{visSet}_{\mathsf{weak}}$ (resp. $\mathsf{visSet}_{\mathsf{strong}}$) will contain the only minimum visibility relation extending $\mathsf{rf}_{\mathsf{weak}}$ (resp. $\mathsf{rf}_{\mathsf{strong}}$) , i.e $\mathsf{vis}^{\min}_{\mathsf{weak}}$ (resp. $\mathsf{vis}^{\min}_{\mathsf{strong}}$.).

For each pair of visibility relations from $\mathsf{visSet}_{\mathsf{weak}}$ and $\mathsf{visSet}_{\mathsf{strong}}$ we compute their stable extensions $\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}}$ which individually satisfy $VisBasic(\alpha_w)$ and $VisBasic(\alpha_s)$, respectively, and jointly satisfy ϕ (line 91). We then check if this computed extension has a bad pattern (Line 93). If no bad patterns are found, we return the $(\mathsf{rf}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}})$ as the witness.

If none of the rf can be extended to obtain the required visibility relation, we declare that the history is a bad history.

Theorem 13 (Correctness of TestMultiLevelCorrectness procedure). *For a hybrid read-write history $H = (\mathcal{O}, \mathsf{so})$ with weak and strong consistency criteria α_w and α_s respectively and multilevel constraints given by ϕ, the procedure* TestMultiLevelCorrectness *returns a witness $(\mathsf{rf}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}})$ over H iff H is multi-level correct with respect to α_w, α_s and ϕ*

Proof. In Appendix B.

4.3 Complexity

Suppose $H = (\mathcal{O}, \mathsf{so})$ is history with $|\mathcal{O}| = N$.

We note that in the procedure ComputeStableExtension, at the end of every iteration of the outer **while**-loop, the values of $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ monotonically increase from the end of the previous iteration. Since they are binary relations over finite history $H = (\mathcal{O}, \mathsf{so})$ their size is upper bounded by $O(N^2)$. The time taken to evaluate each term in $RelTerms(\alpha_\ell)$ is again polynomial in N. Hence, the time-complexity of ComputeStableExtension is polynomial in N, say $f(N)$.

We can observe from the procedure TestMultiLevelCorrectness that the main part that adds to the complexity is iterating through all the Reads-from relation and the total orders if α_w or α_s contain the $\mathsf{totalvis}$ subformula. Suppose the number of read operations are k. Then the number of write operations is $N - k$, and there are $O((N-k)^k)$-many reads-from relations. Since $k = O(N)$, this can be bound by $O(2^{N \log N})$. Furthermore, for a given rf, if any of the levels $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$ require that the visibility relation be a total order, then we iterate over all the total-orders containing the minimal visibility relation extending rf. Iterating through this requires time bounded by $O(2^{N \log N})$. Thus the worst case time complexity of the procedure is $O(f(N).2^{N \log N})$.

In general, the problem of testing the correctness of a hybrid history is in NP. We need to guess the reads-from relation, and then, extend it to obtain the minimal visibility relations satisfying the visibility constraints of the weak and the

strong consistency criteria. If the visibility relation is required to be a total order, we can guess the order. Extending this to derive a fixed-point minimal visibility relations that satisfy all the visibility constraints via ComputeStableExtension requires polynomial time. Subsequently checking for each of the Bad-Patterns requires polynomial time.

Note that we can reduce the testing of the correctness of a non-hybrid regular history with respect to consistency criteria α to this procedure by defining the level of all the read operations to strong. We set α_s to α, α_w to $\top$, and ϕ to ψ^{ext}_{strong}. For any reads-from relation rf, $\mathsf{rf}_{\mathsf{weak}} = \emptyset$. Thus $\mathsf{vis}_{\mathsf{weak}} = \emptyset$, trivially satisfying α_w as well as ψ^{ext}_{strong}. Thus, the lower bound for testing the correctness of the hybrid history is the complexity of testing the correctness of the component individual history fragments with respect to their respective consistency criteria. It has been shown in [9] that testing the correctness of a read-write history with respect to Sequential consistency is NP-COMPLETE. In [3], the authors use the same reduction to show that testing the correctness with respect to Causal Consistency is NP-COMPLETE. However, it can be shown that the reduction works for any consistency criteria stronger than FIFO consistency, and checking correctness with respect to such a consistency criteria is NP-COMPLETE. Thus, in general the problem of testing the Multi-level correctness of a Hybrid History is a hard-problem, but the hardness comes not due to the multilevel constraints but due to the constraints of the individual consistency criteria and the read-write specification.

In [3], the authors identify a class of read-write data-stores called *data-independent* data-stores whose behaviour is not dependent on the exact values written to the keys of the store. Thus, for these stores, if there is a bad history, there is an equivalent bad *differentiated history* where a particular value is written to a particular memory location at most once.Thus, for such data-stores, we can restrict our testing to only the correctness of differentiated histories. The authors show that the problem of testing the correctness of differentiated-histories with respect to Causal Consistency is solvable in Polynomial time.

Note that for differentiated histories, there is exactly one Reads-From relation which associates every Read operation with atmost one Write operation which has written that value to the memory location read by the Read operation. Thus, if neither of α_w or α_s contain total(`vis`) subformula, then, we can see that the procedure TestMultiLevelCorrectness terminates in polynomial time. Thus, our procedure generalizes the result from [3] to all the consistency criteria defined by the grammar, which don't require the visibility relation to be a total order. Moreover, our procedure checks the multi-level correctness of hybrid histories where the individual consistency levels don't require the visibility relation to be a total order, in polynomial time.

On the other hand, if one of α_w or α_s contains total(`vis`), then the worst case complexity remains $O(2^{N \log N})$. Once again, this does not come as a surprise, since the problem of testing the correctness of a differentiated history with respect to sequential consistency is not known to have a polynomial time solution.

5 Related Work

There is prior work in the literature that illustrate the need for multiple levels of consistency provided by the distributed data-stores to provide a trade off between consistency and availability/latency [1, 12, 13, 15]. The work by Kraska et al. [13] provides a transactional paradigm that allows the applications to define the consistency level on the data instead of the transaction, and also allows the application to switch consistency guarantees at runtime. In the work by Guerraoui et al. [12], the authors provide a generic library that allows the applications to request multiple responses to the same query, where the response that comes later in time is *more-correct* than the prior responses. Thus the subsequent responses are supposed to have more knowledge of the state of the system compared to the prior responses. In our work, we have defined multilevel constraints, which will model the requirement of incremental consistency guarantees by requiring that subsequent strong responses see the effects observed the prior weak responses.

Burckhardt in his book [5] provides a generic way for formalizing the specification of distributed data-stores in terms of histories, visibility and arbitration orders and provides an axiomatic characterization for consistency criteria. In our work, we have derived the specification for read-write stores based on the formalism in this book and have following the axiomatic characterization for the consistency criteria. We have provided a grammar that will generate the consistency criteria as a conjunction of individual axioms. Our work extends [5] in terms of the definition of Hybrid Histories and provides a definition of multi-level correctness for read-write stores.

There is prior work on verifying the correctness of a behaviour with respect to individual consistency criteria. Example include [4] which deals with verifying the correctness with respect to eventual consistency, [2] which investigates the feasibility of checking concurrent implementation with respect to consistency criteria that has a sequential specification, including sequential consistency, linearizability and conflict-serializability and [3] which focusses on correctness with respect to Causal Consistency. Our work provides a generic procedure for checking the correctness of read-write histories for all these individual consistency criteria. Further, [3] show that verification of correctness of a history with respect to Causal Consistency is NP-Complete. However, for differentiated histories, the problem is solvable in Polynomial time. In our work, we generalize the technique of computing the minimal visibility relation and checking for the absence of bad patterns for all the consistency criteria defined using the grammar. In [9] we have a detailed complexity analysis of the problem of testing the correctness of a history with respect to various consistency criteria. Our findings in this paper, are consistent with the results from [9] with respect to hardness of testing on consistency criteria that require the visiblity relation to be a total order. In a recent work [8], the authors provide a technique for testing the correctness of a history of a data-store with respect to a weak consistency criteria. That work also characterizes correctness in terms of minimal visibility relation extending the session order (called program-order there) and the happened-before relation (called as

returns-before relation in [5]). Our work applies this concept to read-write stores, where we observe that correctness with respect to visibility constraints can be satisfied by constructing a minimal visibility relation while the correctness with respect to read-write specifications and arbitration constraints can be reduced to checking for absence of certain bad patterns. In particular, our characterization of the arbitration relation in terms of the conflict relation saves the step of searching through all possible arbitration relations which the [8] work does.

[11] deals with verification of *red-blue* consistency where in a history a subset of operations are labelled as red while the remaining are labelled as blue. The blue operations are expected to satisfy weaker consistency criteria, while the subset of red operations are supposed to satisfy a stronger consistency criteria. The effects of the strong operation and weak operations are visible to each other. We can model this by setting $\phi = \psi^{ext}_{strong} \wedge \psi^{ext}_{weak}$.

References

[1] Bailis, P., Ghodsi, A., Hellerstein, J.M., Stoica, I.: Bolt-on causal consistency. In: Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data. pp. 761–772. SIGMOD '13, ACM, New York, NY, USA (2013), http://doi.acm.org/10.1145/2463676.2465279

[2] Bouajjani, A., Emmi, M.: Analysis of recursively parallel programs. ACM Trans. Program. Lang. Syst. 35(3), 10:1–10:49 (2013), https://doi.org/10.1145/2518188

[3] Bouajjani, A., Enea, C., Guerraoui, R., Hamza, J.: On verifying causal consistency. In: Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages. pp. 626–638. POPL 2017, ACM, New York, NY, USA (2017), http://doi.acm.org/10.1145/3009837.3009888

[4] Bouajjani, A., Enea, C., Hamza, J.: Verifying eventual consistency of optimistic replication systems. In: The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014. pp. 285–296 (2014), https://doi.org/10.1145/2535838.2535877

[5] Burckhardt, S.: Principles of eventual consistency. Foundations and Trends in Programming Languages 1(1-2), 1–150 (2014), https://doi.org/10.1561/2500000011

[6] Burkhardt, S., Gotsman, A., Yang, H., Zawirski, M.: Replicated data types: specification, verification, optimality. In: The 41st Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL '14, San Diego, CA, USA, January 20-21, 2014. pp. 271–284 (2014)

[7] Damien.: DynamoDB vs Cassandra (2017 (Accessed Nov 16, 2018)), https://www.beyondthelines.net/databases/dynamodb-vs-cassandra/

[8] Emmi, M., Enea, C.: Monitoring weak consistency. In: Computer Aided Verification - 30th International Conference, CAV 2018, Held as Part of the Federated Logic Conference, FloC 2018, Oxford, UK, July 14-17, 2018, Proceedings, Part I. pp. 487–506 (2018), https://doi.org/10.1007/978-3-319-96145-3_26

[9] Furbach, F., Meyer, R., Schneider, K., Senftleben, M.: Memory model-aware testing - A unified complexity analysis. In: 14th International Conference on Application of Concurrency to System Design, ACSD 2014, Tunis La Marsa, Tunisia, June 23-27, 2014. pp. 92–101 (2014), https://doi.org/10.1109/ACSD.2014.27

[10] Gilbert, S., Lynch, N.A.: Brewer's conjecture and the feasibility of consistent, available, partition-tolerant web services. SIGACT News 33(2), 51–59 (2002)
[11] Gotsman, A., Yang, H., Ferreira, C., Najafzadeh, M., Shapiro, M.: 'cause i'm strong enough: reasoning about consistency choices in distributed systems. In: Proceedings of the 43rd Annual ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, POPL 2016, St. Petersburg, FL, USA, January 20 - 22, 2016. pp. 371–384 (2016), https://doi.org/10.1145/2837614.2837625
[12] Guerraoui, R., Pavlovic, M., Seredinschi, D.A.: Incremental consistency guarantees for replicated objects. In: Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. pp. 169–184. OSDI'16, USENIX Association, Berkeley, CA, USA (2016), http://dl.acm.org/citation.cfm?id=3026877.3026891
[13] Kraska, T., Hentschel, M., Alonso, G., Kossmann, D.: Consistency rationing in the cloud: Pay only when it matters. PVLDB 2(1), 253–264 (2009), http://www.vldb.org/pvldb/2/vldb09-759.pdf
[14] Lamport, L.: How to make a multiprocessor computer that correctly executes multiprocess programs. IEEE Trans. Comput. 28(9), 690–691 (Sep 1979)
[15] Li, C., Porto, D., Clement, A., Gehrke, J., Preguiça, N., Rodrigues, R.: Making geo-replicated systems fast as possible, consistent when necessary. In: Proceedings of the 10th USENIX Conference on Operating Systems Design and Implementation. pp. 265–278. OSDI'12, USENIX Association, Berkeley, CA, USA (2012), http://dl.acm.org/citation.cfm?id=2387880.2387906
[16] Ozkan, B.K., Majumdar, R., Niksic, F., Befrouei, M.T., Weissenbacher, G.: Randomized testing of distributed systems with probabilistic guarantees. PACMPL 2(OOPSLA), 160:1–160:28 (2018), http://doi.acm.org/10.1145/3276530
[17] Perrin, M., Mostefaoui, A., Jard, C.: Causal consistency: Beyond memory. In: Proceedings of the 21st ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming. pp. 26:1–26:12. PPoPP '16, ACM, New York, NY, USA (2016)
[18] Terry, D.B., Theimer, M., Petersen, K., Demers, A.J., Spreitzer, M., Hauser, C.: Managing update conflicts in bayou, a weakly connected replicated storage system. In: Proceedings of the Fifteenth ACM Symposium on Operating System Principles, SOSP 1995, Copper Mountain Resort, Colorado, USA, December 3-6, 1995. pp. 172–183 (1995), https://doi.org/10.1145/224056.224070
[19] Wolper, P.: Expressing interesting properties of programs in propositional temporal logic. In: Conference Record of the Thirteenth Annual ACM Symposium on Principles of Programming Languages, St. Petersburg Beach, Florida, USA, January 1986. pp. 184–193 (1986), https://doi.org/10.1145/512644.512661

A Correctness of the Bad Patterns Charecterization

Lemma 14. *If* rf_ℓ *is a reads-from relation over the fragment* H_ℓ *and* $(\mathsf{vis}_\ell, \mathsf{arb})$ *realize* rf_ℓ*. Then,* $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \subseteq \mathsf{arb}$.

Proof. Suppose $(o'', o') \in \mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell)$. By definition, there exists a Read operation o such that both o', o'' are in the maximal related writes of o and $o' = \mathsf{rf}^{-1}(o)$. Since rf is realized by $(\mathsf{vis}_\ell, \mathsf{arb})$, by definition, $\mathsf{rf}^{-1}(o) = \mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(()o)$. Hence o' is the effective write of o.

Now by the definition, an effective write of a read operation is arbitrated after all the remaining maximal related writes of that read operation. Thus $(o'', o) \in \mathsf{arb}$.

With this, we prove the correctness of Theorem 12

Given a hybrid history $H = (\mathcal{O}, \mathsf{so})$ with weak and strong consistency criteria defined by $\alpha_w = \beta_w \wedge \gamma_w$ and $\alpha_s = \beta_s \wedge \gamma_s$ respectively. Let the multilevel constraints be defined by ϕ. We need to show that H is multilevel correct with respect to α_w, α_s and ϕ iff there exists a reads-from relation rf and visibility relations $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ that extend $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ respectively such that $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$, $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$ and none of the bad patterns in
$\{\mathsf{BADVISIBILITY}, \mathsf{THINAIR}, \mathsf{BADINITREAD}, \mathsf{BADREAD}, \mathsf{BADARB}\}$ exists.

Proof. ($\Longrightarrow$): Suppose hybrid history H is correct. Then, there exists visibilty relations $\mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}}$ and arbitration relations arb such that $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{arb} \models \alpha_w$, $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}}, \mathsf{arb} \models \alpha_s$ and $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$.

Since $\alpha_w = \beta_w \wedge \gamma_w$, we have $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$ and $(\mathsf{vis}_{\mathsf{weak}})_{\mathsf{Write}} \subseteq \mathsf{arb}$ since $\gamma_w = \gamma_s = (\mathtt{vis})_{\mathsf{Write}} \subseteq \mathtt{arb}$. By similar reasoning we have $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$ and $(\mathsf{vis}_{\mathsf{strong}})_{\mathsf{Write}} \subseteq \mathsf{arb}$

For $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$,we set $\mathsf{rf}_\ell = \{(\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o), o) \mid o \in \mathcal{O} \wedge \mathit{Op}(o) = \mathsf{Read} \wedge \mathit{Level}(o) = \ell\}$. $\mathsf{rf} = \mathsf{rf}_{\mathsf{weak}} \cup \mathsf{rf}_{\mathsf{strong}}$. By definition $\mathsf{vis}_{\mathsf{weak}}$ extends $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ extends $\mathsf{rf}_{\mathsf{strong}}$.

We will now show that none of the aforementioned bad patterns exists for H, rf, $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$.

Since H is multilevel correct, $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ by definitions are acyclic relations. So $\mathsf{BADVISIBILITY}$ bad pattern doesn't exist.

Further, due to correctness of H, by the definition of $\mathsf{Spec}_{\mathsf{RW}}$, for any read operation o of level ℓ, $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = \bot$ iff $\mathit{Ret}(o) = 0$. Since for every read $\mathsf{rf}^{-1}(o) = \mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o)$, $\mathsf{THINAIR}$ bad pattern doesn't exist.

Since $H_\ell, \mathsf{vis}_\ell, \mathsf{arb}$ is correct with respect to $\mathsf{Spec}_{\mathsf{RW}}$, for any read operation o with level ℓ such that $\mathit{Ret}(o) = 0$, $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = \bot$ which implies that $\mathit{RelWrites}_{\mathsf{vis}_\ell}(o) = \emptyset$. Hence, $\mathsf{BADINITREAD}$ bad pattern doesn't exist.

For a correct history H, for any read operation o with level ℓ, if $\mathit{Ret}(o) \neq 0$, then $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) \in \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$. Since $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = \mathsf{rf}^{-1}(o)$, $\mathsf{BADREAD}$ bad pattern doesn't exist.

By lemma 14 for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$, $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \subseteq \mathsf{arb}$. Since arb is consistent with both $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$, we have $(\mathsf{vis}_\ell)_{\mathsf{Write}} \subseteq \mathsf{arb}$. Hence

$\bigcup_{\ell \in \{\mathsf{weak},\mathsf{strong}\}} (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \cup (\mathsf{vis}_\ell)_{\mathsf{Write}}) \subseteq \mathsf{arb}$. By definition arb is a total order. Thus BADARB would imply a cycle in arb which is not true. Hence BADARB bad pattern doesn't exist.

($\Longleftarrow$): Suppose there exists a rf and $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ such that $H, \mathsf{vis}_{\mathsf{weak}} \models \beta_w$ and $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \beta_s$ and $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$. We will now construct an arbitration relation arb.

Since the BADARB bad pattern doesn't exist, $\bigcup_{\ell \in \{\mathsf{weak},\mathsf{strong}\}} (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \cup (\mathsf{vis}_\ell)_{\mathsf{Write}})$ is an acyclic relation. We set arb to be a topological sort of this acyclic relation along with the Write operations from o, not appearing in this acyclic relations. Thus arb is a total order. By construction, $(\mathsf{vis}_\ell)_{\mathsf{Write}} \subseteq \mathsf{arb}$ for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$. Thus, $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{arb} \models \gamma_w$ and $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}}, \mathsf{arb} \models \gamma_s$. From this, and what is given we can conclude that $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{arb} \models \alpha_w$ and $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}}, \mathsf{arb} \models \alpha_s$.

We now only need to show that for each $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$, $H_\ell, \mathsf{vis}_\ell, \mathsf{arb} \models \mathsf{Spec}_{\mathsf{RW}}$.

Let o be a read operation with level ℓ. Suppose $\mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o) = \emptyset$. Then $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = \bot$. Since $\mathsf{rf}_\ell \subseteq \mathsf{vis}_\ell$, $\mathsf{rf}_\ell^{-1}(o) = \bot$. Since THINAIR bad pattern doesn't exist, it has to be the case that $\mathit{Ret}(o) = 0$. Thus, if $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = \bot$ then $\mathit{Ret}(o) = 0$. Conversely, suppose $\mathit{Ret}(o) = 0$. Then, since BADINITREAD bad pattern doesn't exist, $\mathit{RelWrites}_{\mathsf{vis}_\ell}(o) = \emptyset$. Thus, by definition, $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}\ell}(o) = \bot$. Thus, we can conclude that $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}\ell}(o) = \bot \iff \mathit{Ret}(o) = 0$.

Suppose $\mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o) \neq \emptyset$. Since BADINITREAD bad pattern doesn't exist, $\mathit{Ret}(o) \neq 0$. Further, since THINAIR badpattern doesn't exist, $\mathsf{rf}_\ell^{-1}(o) \neq \bot$. Let $\mathsf{rf}_\ell^{-1}(o) = o'$. Since BADREAD bad pattern doesn't exist, $\mathsf{rf}_\ell^{-1}(o) = o' \in \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$. For any $o'' \in \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$ we have $(o'', o') \in \mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell)$. Now, by construction of arb, we have $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \subseteq \mathsf{arb}$.

Thus, for any $o'' \in \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$, $o'' \xrightarrow{\mathsf{arb}} o'$. Thus by definition, $\mathit{EffWrite}^{\mathsf{arb}}_{\mathsf{vis}_\ell}(o) = o'$. However, since $o' = \mathsf{rf}_\ell^{-1}(o)$ by definition of a reads-from relation $\mathit{Ret}(o) = \mathit{Args}(o')$.

Since o is an arbitrary Read operation with level ℓ in H, what we have shown holds for all Read operation with level ℓ. Hence $H_\ell, \mathsf{vis}_\ell, \mathsf{arb} \models \mathsf{Spec}_{\mathsf{RW}}$.

B Correctness of the testing procedure

We will first prove a set of lemmas with respect to the termination and the correctness of the helper procedures.

Lemma 15 (Termination of Helper functions). *For a given hybrid history, and a given visibility relations over the history, the methods* BuildMinVisSingle, BuildMinVisMulti *and* ComputeStableExtension *terminate.*

Proof. We first observe that BuildMinVisMulti terminates since it doesn't have any loops. And then visibility relations it outputs is a superset of the input visibility relations.

We will now show the termination of BuildMinVisSingle. Let $\mathsf{vis}_{\mathsf{new}}^{i,j}$ denote the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of the j^{th} iteration of the inner **for**-loop within the i^{th} iteration of the outer **while**-loop. Let $\mathsf{vis}_{\mathsf{new}}^{i}$ denote the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of the i^{th} iteration of the outer **while**-loop.

We note that for $j > 0$, $\mathsf{vis}_{\mathsf{new}}^{i,j} \supseteq \mathsf{vis}_{\mathsf{new}}^{i,j-1}$ since we only keep extending $\mathsf{vis}_{\mathsf{new}}$ inside the inner **for**-loop by adding to it the result of evaluation of the *RelTerms* in α_ℓ. $\mathsf{vis}_{\mathsf{new}}^{i,0} = \mathsf{vis}_{\mathsf{new}}^{i-1}$. If $|RelTerms(\alpha_\ell)| = n$, then, $\mathsf{vis}_{\mathsf{new}}^{i} = \mathsf{vis}_{\mathsf{new}}^{i,n}$. Thus, $\mathsf{vis}_{\mathsf{new}}^{i} \supseteq \mathsf{vis}_{\mathsf{new}}^{i-1}$. At the end of the outer-while loop we check if $\mathsf{vis}_{\mathsf{new}} = \mathsf{vis}_{\mathsf{old}}$ which is equivalent to checking $\mathsf{vis}_{\mathsf{new}}^{i} = \mathsf{vis}_{\mathsf{new}}^{i-1}$. If true, the function returns. Since $\mathsf{vis}_{\mathsf{new}} \subseteq \mathcal{O} \times \mathcal{O}$, and since $\mathcal{O}$ is a finite set, it will be the case that $\mathsf{vis}_{\mathsf{new}} = \mathsf{vis}_{\mathsf{old}}$ after a finite number of iterations. Hence the procedure terminates.

In case of ComputeStableExtension , we note that it obtains the new values for $\mathsf{vis}_{\mathsf{weak}}^{\mathsf{new}}$ and $\mathsf{vis}_{\mathsf{strong}}^{\mathsf{new}}$ individually by invoking the procedure BuildMinVisSingle, which returns a relation that is a superset of the input visibility relation. Similarly, in the inner **for**-loop, we obtain the new values for the pair $(\mathsf{vis}_{\mathsf{weak}}^{\mathsf{new}}, \mathsf{vis}_{\mathsf{strong}}^{\mathsf{new}})$ by calling BuildMinVisMulti, which returns visibility relations that are supersets of the corresponding input relations. Thus, at the end of each iteration of **while**-loop, either the values of $\mathsf{vis}_{\mathsf{weak}}^{\mathsf{new}}$ and $\mathsf{vis}_{\mathsf{strong}}^{\mathsf{new}}$ are the same as their values at the end of the previous iteration of the **while**-loop, or they are a superset of their values at the end of the the previous generation. Since both $\mathsf{vis}_{\mathsf{weak}}^{\mathsf{new}}$ and $\mathsf{vis}_{\mathsf{weak}}^{\mathsf{new}}$ are binary relations over $\mathcal{O}_{\mathsf{weak}}$ and $\mathcal{O}_{\mathsf{strong}}$, their maximal size is bound by $|\mathcal{O}|^2$. Thus, the iterations of the outer while loop are bounded by $O(|\mathcal{O}|^2)$ iterations. Hence ComputeStableExtension terminates

Theorem 16 (Termination of Testing Procedure). *For any given hybrid-history H, and consistency criteria α_w, α_s and multilevel constraints ϕ, the procedure* TestMultiLevelCorrectness *terminates.*

Proof. Since H is a finite history, the number of Reads-From relations that can be defined over it are finite. Further, for each rf from the set of reads-from relations, the extensions $\mathsf{vis}_{\mathsf{weak}}^{\min}$ and $\mathsf{vis}_{\mathsf{strong}}^{\min}$ are finite. In the worst case when both α_w as well as α_s contain the subsformula total(vis), the sizes of $\mathsf{visSet}_{\mathsf{weak}}$ and $\mathsf{visSet}_{\mathsf{strong}}$ is finite. Since the procedures called within the inner **for**-loop, i.e. ComputeStableExtension and CheckBadPatterns, terminate, the inner **for**-loop (Lines 90-94) will iterate only for a finite number of times.

Thus, the procedure will terminate when either it has found a witness rf, $\mathsf{vis}_{\mathsf{weak}}^{\mathsf{stable}}$ and $\mathsf{vis}_{\mathsf{strong}}^{\mathsf{stable}}$ for the correctness of the Hybrid history, or when it has iterated over all the finitely many reads-from relation.

Lemma 17 (Correctness of BuildMinVisSingle**).** *Let vis_ℓ be a visiblity relation over some fragment of a hybrid history H_ℓ. Let α_ℓ be a consistency criteria. Let* $\mathsf{vis} := \mathsf{BuildMinVisSingle}(H_\ell, \mathsf{vis}_\ell, \alpha_\ell)$. *Then $H_\ell, \mathsf{vis} \models VisBasic(\alpha_\ell)$.*

Proof. We will denote the value at the end of the i^{th} iteration of the outer while-loop as $\mathsf{vis}_{\mathsf{new}}^{i}$. We shall denote the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of the the j^{th} iteration in the i^{th} iteration of the inner for loop as $\mathsf{vis}_{\mathsf{new}}^{i,j}$.

Let vis be the value returned by BuildMinVisSingle at the end of the i^{th} iteration of the outer **while**-loop. Then, $\mathsf{vis} = \mathsf{vis}^{i}_{\mathsf{new}}$.

Note that $\mathsf{vis}_{\mathsf{old}}$ is the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of the previous iteration of while loop. Thus $\mathsf{vis}_{\mathsf{old}} = \mathsf{vis}^{i-1}_{\mathsf{new}}$. Further since $\mathsf{vis}_{\mathsf{old}} = \mathsf{vis}_{\mathsf{new}}$ for the function to return, we have $\mathsf{vis}^{i}_{\mathsf{new}} = \mathsf{vis}^{i-1}_{\mathsf{new}}$. Let $\mathsf{vis}^{i,0}_{\mathsf{new}}$ denote the value of $\mathsf{vis}_{\mathsf{new}}$ at the beginning of the inner for-loop. Then $\mathsf{vis}^{i,0}_{\mathsf{new}} = \mathsf{vis}^{i-1}_{\mathsf{new}}$. Suppose $\mathit{RelTerms}(\alpha)$ has n terms where the k^{th} term is denoted by τ_k , then, we can see that for $j \in [1, \ldots, n]$, $\mathsf{vis}^{i,j}_{\mathsf{new}} = \mathsf{vis}^{i,j-1}_{\mathsf{new}} \cup \tau_j[\mathsf{so}_\ell, \mathsf{vis}^{i,j-1}_{\mathsf{new}}]$. Thus, we can conclude that $\tau_j[\mathsf{so}_\ell, \mathsf{vis}^{i,j-1}_{\mathsf{new}}] \subseteq \mathsf{vis}^{i,j}_{\mathsf{new}}$.

Also, we can note that $\mathsf{vis}^{i-1}_{\mathsf{new}} = \mathsf{vis}^{i,0}_{\mathsf{new}} \subseteq \mathsf{vis}^{i,1}_{\mathsf{new}} \subseteq \cdots \subseteq \mathsf{vis}^{i,n}_{\mathsf{new}} = \mathsf{vis}^{i}_{\mathsf{new}}$. Since, $\mathsf{vis}^{i-1}_{\mathsf{new}} = \mathsf{vis}^{i}_{\mathsf{new}}$, this implies that for each $j \in [0, \ldots, n]$, $\mathsf{vis}^{i,j}_{\mathsf{new}} = \mathsf{vis}^{i}_{\mathsf{new}} = \mathsf{vis}$.

Thus, for each $j \in [1, \ldots, n]$, we have $\tau_j[\mathsf{so}_\ell, \mathsf{vis}] \subseteq \mathsf{vis}$. Hence, $\mathsf{so}_\ell, \mathsf{vis} \models \bigwedge_{\tau_j \in RelTerms(\alpha_\ell)} \tau_j \subseteq \mathtt{vis}$. But by definition, $\bigwedge_{\tau_j \in RelTerms(\alpha_\ell)} \tau_j \subseteq \mathtt{vis} = \mathit{VisBasic}(\alpha)$. Hence $\mathsf{so}_\ell, \mathsf{vis} \models \mathit{VisBasic}(\alpha)$ which implies that $H_\ell, \mathsf{vis} \models \mathit{VisBasic}(\alpha)$.

Lemma 18 (Monotonicity of *RelTerms*).
Let $H = (\mathcal{O}, \mathsf{so})$ be a well defined fragment of a hybrid history Let vis and vis' be two visibility relation over H such that $\mathsf{vis} \subseteq \mathsf{vis}'$. Then for any term $\tau \in \mathit{RelTerms}$, $\tau[\mathsf{so}, \mathsf{vis}] \subseteq \tau[\mathsf{so}, \mathsf{vis}']$.

Proof. We will prove this by induction over the number of compositions in the term τ. The base case is when there are no compositions. We have two cases $\tau = \mathtt{so}$ and $\tau = \mathtt{vis}$.

In the former case, the result trivially follows. In the latter case, the result follows since it is given that $\mathsf{vis} \subseteq \mathsf{vis}'$.

Suppose the result holds for all τ with fewer than n compositions. We now consider a $\tau = \tau'; \tau''$ where both τ' and τ'' have at most $n-1$ compositions. Now $\tau[\mathsf{so}, \mathsf{vis}] = \tau'[\mathsf{so}, \mathsf{vis}]; \tau''[\mathsf{so}, \mathsf{vis}]$. By induction hypothesis, $\tau'[\mathsf{so}, \mathtt{vis}] \subseteq \tau'[\mathsf{so}, \mathsf{vis}']$ and $\tau''[\mathsf{so}, \mathtt{vis}] \subseteq \tau''[\mathsf{so}, \mathsf{vis}']$. Since $A \subseteq A'$ and $B \subseteq B'$ implies $A; B \subseteq A'; B'$ we can conclude that $\tau'[\mathsf{so}, \mathsf{vis}]; \tau''[\mathsf{so}, \mathsf{vis}] \subseteq \tau'[\mathsf{so}, \mathsf{vis}']; \tau''[\mathsf{so}, \mathsf{vis}'] = \tau[\mathsf{so}, \mathsf{vis}']$. Thus the result is true for a τ with n compositions.

Hence, the result is true for all $\tau \in \mathit{RelTerms}$

Lemma 19 (Minimality of BuildMinVisSingle). *Let vis_ℓ be a visiblity relation over some fragment of a hybrid history H_ℓ. Let C_ℓ be axioms defining the consistency criteria. Let vis' be a visibility relation over H_ℓ such that $H_\ell, \mathsf{vis}' \models \mathit{VisBasic}(\alpha_\ell)$.*

Then if, $\mathsf{vis} :=$ BuildMinVisSingle$(H_\ell, \mathsf{vis}_\ell, C_\ell)$, we have $\mathsf{vis} \subseteq \mathsf{vis}'$.

Proof. As before, we will denote the value at the end of the i^{th} iteration of the outer while-loop as $\mathsf{vis}^{i}_{\mathsf{new}}$. We shall denote the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of the the j^{th} iteration in the i^{th} iteration of the inner for loop as $\mathsf{vis}^{i,j}_{\mathsf{new}}$. We set $\mathsf{vis}^{0}_{\mathsf{new}} = \mathsf{vis}^{0,0}_{\mathsf{new}} = \mathsf{vis}_\ell$.

Let $|\mathit{RelTerms}(\alpha)| = n$ and let τ_j denote the j^{th} member of $\mathit{RelTerms}(\alpha)$.

We will first show that for $j \in [1, \ldots, n]$, If $\mathsf{vis}^{i,j-1}_{\mathsf{new}} \subseteq \mathsf{vis}'$ then $\mathsf{vis}^{i,j}_{\mathsf{new}} \subseteq \mathsf{vis}'$. Note that $\mathsf{vis}^{i,j}_{\mathsf{new}} = \mathsf{vis}^{i,j-1}_{\mathsf{new}} \cup \tau_j[\mathsf{so}_\ell, \mathsf{vis}^{i,j-1}_{\mathsf{new}}]$. By assumption, $\mathsf{vis}^{i,j-1}_{\mathsf{new}} \subseteq \mathsf{vis}'$. By lemma 18, $\tau_j[\mathsf{so}_\ell, \mathsf{vis}^{i,j-1}_{\mathsf{new}}] \subseteq \tau_j[\mathsf{so}_\ell, \mathsf{vis}']$. Thus, we can conclude that $\mathsf{vis}^{i,j}_{\mathsf{new}} \subseteq \mathsf{vis}'$.

Since for any i, $\mathsf{vis}^{i,0}_{\mathsf{new}} \subseteq \mathsf{vis}^{i,1}_{\mathsf{new}} \subseteq \cdots \subseteq \mathsf{vis}^{i,n}_{\mathsf{new}} = \mathsf{vis}^{i}_{\mathsf{new}}$, we can conclude that if $\mathsf{vis}^{i,0}_{\mathsf{new}} \subseteq \mathsf{vis}'$ then, $\mathsf{vis}^{i}_{\mathsf{new}} \subseteq \mathsf{vis}'$. Finally note that $\mathsf{vis}^{i}_{\mathsf{new}} = \mathsf{vis}^{i+1,0}_{\mathsf{new}}$. Thus, if $\mathsf{vis}^{i}_{\mathsf{new}} \subseteq \mathsf{vis}'$ then $\mathsf{vis}^{i+1}_{\mathsf{new}} \subseteq \mathsf{vis}'$. Finally we note that $\mathsf{vis}^{0,0}_{\mathsf{new}} = \mathsf{vis}_{\ell} \subseteq \mathsf{vis}'$. Thus for all $i > 0$, $\mathsf{vis}^{i}_{\mathsf{new}} \subseteq \mathsf{vis}'$. Since the value vis returned by BuildMinVisSingle is the value of $\mathsf{vis}_{\mathsf{new}}$ at the end of some iteration i, it follows that $\mathsf{vis} \subseteq \mathsf{vis}'$.

Lemma 20 (Correctness of BuildMinVisMulti). *Let $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ be visibility relations over fragments H_{weak} and H_{strong} of a hybrid history $H = (\mathcal{O}, \mathsf{so})$. Let ψ be a subformula in ϕ.*

Let $(\mathsf{vis}^{ret}_{\mathsf{weak}}, \mathsf{vis}^{ret}_{\mathsf{strong}}) = \mathsf{BuildMinVisMulti}(\mathcal{O}, \mathsf{so}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}}, \psi)$.

Then, $\mathsf{vis}_{\mathsf{weak}} \subseteq \mathsf{vis}^{ret}_{\mathsf{weak}}$, $\mathsf{vis}_{\mathsf{strong}} \subseteq \mathsf{vis}^{ret}_{\mathsf{strong}}$ and $H, \mathsf{vis}^{ret}_{\mathsf{weak}}, \mathsf{vis}^{ret}_{\mathsf{strong}} \models \psi$.

Proof. We shall prove the result for the cases when $\psi = \psi^{ext}_{strong}$, ψ^{mr}_{strong} and ψ^{rest}_{strong}. The remaining cases are symmetric to these.

For each of these cases, we note that $\ell = \mathsf{strong}$ and $\ell' = \mathsf{weak}$ and $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} = \mathsf{vis}_{\mathsf{strong}}$ and $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} = \mathsf{vis}_{\mathsf{weak}}$.

Suppose $\psi = \psi^{ext}_{strong}$. Now $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$. Thus, we can rewrite this as $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}}$. Thus, $(\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$. Hence, we can write that $\mathsf{so}, \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \models \psi^{ext}_{strong}$. The proof follows for this case.

Suppose $\psi = \psi^{mr}_{strong}$. Then, $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}}$, and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$. Now if $(o', o) \in (\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}}$. Then, there exists an operation o_1 such that $(o', o_1) \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ and $(o_1, o) \in \mathsf{so}_{\mathsf{strong}} \subseteq \mathsf{so}$. Thus $o_1 \in \mathcal{O}_{\mathsf{strong}}$.

We consider two subcases. If (o', o_1) was already present in $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ then, since $(\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ it implies that $(o', o) \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$. Suppose (o', o_1) was not originally present in $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$. It implies that $(o', o_1) \in (\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}}$. This implies that there exists an o_2 such that $(o', o_2) \in \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ and $(o_2, o_1) \in \mathsf{so}$. Thus $o_2 \in \mathcal{O}_{\mathsf{strong}}$. Which implies that $(o_2, o_1) \in \mathsf{so}_{\mathsf{strong}}$. We have already shown that $(o_1, o) \in \mathsf{so}_{\mathsf{strong}}$. Thus, $(o_2, o) \in \mathsf{so}_{\mathsf{strong}} \subseteq \mathsf{so}$. Since $(o', o_2) \in \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ and $(o_2, o) \in \mathsf{so}_{\mathsf{strong}} \subseteq \mathsf{so}$, it implies that $(o', o) \in (\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$. Thus, $(\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$. Thus $\mathsf{so}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models (\mathtt{vis}^{\mathtt{strong}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathtt{vis}^{\mathtt{strong}}$. We can conclude that $H, \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \models \psi^{mr}_{strong}$

Suppose $\psi = \psi^{rest}_{strong}$. Now $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} \cup (\mathsf{UnaccountedWrites}; \mathsf{NearestPriorOther})$. We need to show that if $(o', o) \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ then, there should exist an o'' in $\mathcal{O}_{\mathsf{weak}}$ such that $(o', o'') \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $(o'', o) \in \mathsf{so}$.

We consider two subcases. $(o', o) \notin \mathsf{UnaccountedWrites}$. Then, since $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$, and $\mathsf{UnaccountedWrites} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \setminus (\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})$, it implies that $(o', o) \in \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so}$. Since $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$, we have $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}; \mathsf{so}$. This implies that $(o', o) \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}; \mathsf{so}$ which proves this subcase.

Suppose $(o', o) \in \mathsf{UnaccountedWrites}$. We set o'' to be the event such that $o'' \xrightarrow[\mathsf{weak}]{\mathsf{so}(1)} o$. Thus by definition, $(o, o'') \in \mathsf{NearestPriorOther}$.

Which implies that $(o', o'') \in \mathsf{UnaccountedWrites}; \mathsf{NearestPriorOther}$.

Thus, by definition of $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ in this case, $(o', o'') \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$. Since $o'' =\xrightarrow[\mathsf{weak}]{\mathsf{so}(1)}$ (o), $(o'', o) \in \mathsf{so}$. This completes the proof for this subcase.

From this we can conclude that $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}; \mathsf{so}$. Thus $\mathsf{so}, \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \models \mathtt{vis}^{\mathtt{strong}} \subseteq \mathtt{vis}^{\mathtt{weak}}; \mathsf{so}$. Hence $H, \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{strong}}_{\mathsf{weak}} \models \psi^{rest}_{strong}$.

Thus in each of these cases, $H, \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{strong}}_{\mathsf{weak}} \models \phi$. Further, in case when $\psi = \psi^{ext}_{strong}$ or ψ^{mr}_{strong}, $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \supseteq \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$. Similarly in case when $\psi = \psi^{rest}_{strong}$, $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} \supseteq \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$. Since $\mathsf{vis}^{ret}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $\mathsf{vis}^{ret}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$, the proof of this lemma is complete.

Lemma 21 (Minimality of BuildMinVisMulti**).** *Let* $\mathsf{vis}_{\mathsf{weak}}$ *and* $\mathsf{vis}_{\mathsf{strong}}$ *be visibility relations over fragments* H_{weak} *and* H_{strong} *of a hybrid history* $H = (\mathcal{O}, \mathsf{so})$. *Let* ψ *be a subformula in the hybrid constraint* ϕ.

Suppose there exists $\mathsf{vis}'_{\mathsf{weak}}$ *and* $\mathsf{vis}'_{\mathsf{strong}}$ *over* H_{weak} *and* H_{strong} *respectively such that*

- $\mathsf{vis}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$
- $\mathsf{vis}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$
- $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \psi$.

Then, if $(\mathsf{vis}^{ret}_{\mathsf{weak}}, \mathsf{vis}^{ret}_{\mathsf{strong}}) = \mathsf{BuildMinVisMulti}(\mathcal{O}, \mathsf{so}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}}, \phi)$, *it is the case that* $\mathsf{vis}^{ret}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ *and* $\mathsf{vis}^{ret}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$

Proof. We have $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} = \mathsf{vis}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} = \mathsf{vis}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$ We will prove the result for the case when ψ is one of ψ^{ext}_{strong}, ψ^{mr}_{strong} or ψ^{rest}_{strong} the other cases follow in a similar fashion.

Suppose ψ is ψ^{ext}_{strong}.

Since $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$, we have $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so} \subseteq \mathsf{vis}'_{\mathsf{weak}}; \mathsf{so}$. From this, we have $(\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq (\mathsf{vis}'_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}}$. This implies $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}} \cup (\mathsf{vis}'_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}}$ since $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$. Since $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \psi^{ext}_{strong}$ implies $(\mathsf{vis}'_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$ we can conclude that $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$. However $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \cup (\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}; \mathsf{so})_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$. Thus $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$. Hence this case is proved.

The argument for $\psi = \psi^{mr}_{strong}$ is same as above with $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$ replaced with $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$, $\mathsf{vis}'_{\mathsf{weak}}$ replaced by $\mathsf{vis}'_{\mathsf{strong}}$.

Suppose ψ is ψ^{rest}_{strong}. Then, $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$.

$\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}} \cup \mathsf{UnaccountedWrites}; \mathsf{NearestPriorOther}$. We need to prove that if $(o', o) \in \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ then, $(o', o) \in \mathsf{vis}'_{\mathsf{weak}}$.

Now,if $(o', o) \in \mathsf{vis}^{\mathsf{old}}_{weak}$, since $\mathsf{vis}^{\mathsf{old}}_{weak} \subseteq \mathsf{vis}'_{\mathsf{weak}}$, the proof follows.

Suppose $(o', o) \notin \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$. Then, $(o', o) \in \mathsf{UnaccountedWrites}; \mathsf{NearestPriorOther}$. This implies that there exists an $\mathcal{O}_{\mathsf{strong}}$ operation o'' such that $(o', o'') \in \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$ and $o =\xrightarrow[\mathsf{weak}]{\mathsf{so}(1)} (o'')$. Since $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$, $(o', o'') \in \mathsf{vis}'_{\mathsf{strong}}$.

But then, $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \psi^{rest}_{strong}$. Thus, the write o' should be visible to some $\mathcal{O}_{\mathsf{weak}}$ operation o''' preceding o'' in its session. Thus, $o' \xrightarrow{\mathsf{vis}'_{\mathsf{weak}}} o''' \xrightarrow{\mathsf{so}} o''$.

Since o is the nearest $\mathcal{O}_{\mathsf{weak}}$-predecessor of o'' in its session, we have $o' \xrightarrow{\mathsf{vis}'_{\mathsf{weak}}} o''' \xrightarrow{\mathsf{so}} o \xrightarrow[\mathsf{weak}]{\mathsf{so}} o''$. Further, since since ϕ is well defined, if ψ^{rest}_{strong} is a subformula, then ψ^{mr}_{weak} is also a subformula, which implies that $(\mathsf{vis}'_{\mathsf{weak}}; \mathsf{so})_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$. Thus $o' \xrightarrow{\mathsf{vis}'_{\mathsf{weak}}} o''' \xrightarrow{\mathsf{so}} o$ implies $(o', o) \in \mathsf{vis}'_{\mathsf{weak}}$. This completes the proof for this case.

Lemma 22 (Correctness of ComputeStableExtension). *Let* $\mathsf{vis}_{\mathsf{weak}}$ *and* $\mathsf{vis}_{\mathsf{strong}}$ *be a visibility relation over the fragments* H_{weak} *and* H_{strong} *of a hybrid history* H*. Let* α_w *and* α_s *respectively be the weak and strong consistency criteria and let* ϕ *be the multilevel constraints. Let* $(\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}})$ *be the return value obtained from* $\mathsf{ComputeStableExtension}(\mathcal{O}, \mathsf{so}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}}, \alpha_w, \alpha_s, \phi)$*. Then*

- $H_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{weak}} \models \mathit{VisBasic}(\alpha_w)$
- $H_{\mathsf{strong}}, \mathsf{vis}_{\mathsf{strong}} \models \mathit{VisBasic}(\alpha_s)$
- $H, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}} \models \phi$

Proof. We note that the value returned by ComputeStableExtension is the values of variables $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ at the end of the outer while loop, when they respectively match the values $\mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$, which were the values of $\mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$ at the end of the previous iteration of the outer **while**-loop.

We will replay the iteration of the outer-while loop which returned the value. Here, we note that $\mathsf{vis}^{\mathsf{prev}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$ and $\mathsf{vis}^{\mathsf{prev}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$.

Let the value computed in line 55 by invoking the method BuildMinVisSingle be denoted as $\mathsf{vis}^1_{\mathsf{weak}}$. Now $\mathsf{vis}^1_{\mathsf{weak}} \subseteq \mathsf{vis}^{\mathsf{prev}}_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{weak}}$. Further, by Lemma 17 $H_{\mathsf{weak}}, \mathsf{so}_{\mathsf{weak}}, \mathsf{vis}^1_{\mathsf{weak}} \models \mathit{VisBasic}(\alpha_w)$.

Let the value computed in line 57 by invoking the method BuildMinVisSingle be denoted as $\mathsf{vis}^1_{\mathsf{strong}}$. Now $\mathsf{vis}^1_{\mathsf{strong}} \subseteq \mathsf{vis}^{\mathsf{prev}}_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{old}}_{\mathsf{strong}}$.
Further, $H_{\mathsf{strong}}, \mathsf{so}_{\mathsf{strong}}, \mathsf{vis}^1_{\mathsf{strong}} \models \mathit{VisBasic}(\alpha_s)$.

Let $\phi = \psi_2 \wedge \ldots, \wedge \psi_k$.

We let $(\mathsf{vis}^i_{\mathsf{weak}}, \mathsf{vis}^i_{\mathsf{strong}}) = \mathsf{BuildMinVisMulti}(\mathcal{O}, \mathsf{so}, \mathsf{vis}^{i-1}_{\mathsf{weak}}, \mathsf{vis}^{i-1}_{\mathsf{strong}}, \psi_i)$ for $i \in [2, \ldots, k]$

By lemma 20, for $i \in [2, \ldots, k]$, we have

- $\mathsf{vis}^{i-1}_{\mathsf{weak}} \subseteq \mathsf{vis}^i_{\mathsf{weak}}$
- $\mathsf{vis}^{i-1}_{\mathsf{strong}} \subseteq \mathsf{vis}^i_{\mathsf{strong}}$
- $H, \mathsf{vis}^i_{\mathsf{weak}}, \mathsf{vis}^i_{\mathsf{strong}} \models \psi_i$

And $\mathsf{vis}^k_{\mathsf{weak}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{weak}}$, $\mathsf{vis}^k_{\mathsf{strong}} = \mathsf{vis}^{\mathsf{new}}_{\mathsf{strong}}$

Thus, for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$ we have $\mathsf{vis}^{\mathsf{old}}_\ell \subseteq \mathsf{vis}^1_\ell \subseteq \mathsf{vis}^2_\ell \subseteq \ldots \mathsf{vis}^{\mathsf{new}}_\ell = \mathsf{vis}^{\mathsf{old}}_\ell$.

From this we can conclude that each of $\mathsf{vis}^i_\ell = \mathsf{vis}^{\mathsf{new}}_\ell$ for $i \in [1, \ldots, k]$. This proves the result.

Lemma 23 (Minimality of ComputeStableExtension). *Let* $\mathsf{vis}_{\mathsf{weak}}$ *and* $\mathsf{vis}_{\mathsf{strong}}$ *be a visibility relation over the fragments* H_{weak} *and* H_{strong} *of a hybrid history* H*. Let* α_w*,* α_s *be weak and strong consistency criteria and let* ϕ *be multilevel*

constraints.Let
$(\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}}) := \mathsf{ComputeStableExtension}(\mathcal{O}, \mathsf{so}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}}, \alpha_w, \alpha_s, \phi)$. *If there exists visibility relations* $\mathsf{vis}'_{\mathsf{weak}}$ *and* $\mathsf{vis}'_{\mathsf{strong}}$ *over* H_{weak} *and* H_{strong} *respectively such that Then*

- $\mathsf{vis}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$
- $\mathsf{vis}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$
- $H_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{weak}} \models \mathit{VisBasic}(\alpha_w)$
- $H_{\mathsf{strong}}, \mathsf{vis}'_{\mathsf{strong}} \models \mathit{VisBasic}(\alpha_s)$
- $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \phi$

Then $\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ *and* $\mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$

Proof. The proof for this follows the line of argument showing the minimality of BuildMinVisSingle. We note that at each step we compute extensions of the weak and strong visibility relations via invoking BuildMinVisSingle and BuildMinVisMulti.

From Lemmas 19 and 21, the output produced by these procedures $\mathsf{vis}^{ret}_{\mathsf{weak}}$ and $\mathsf{vis}^{ret}_{\mathsf{strong}}$ from inputs $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ respectively will satisfy $\mathsf{vis}^{ret}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}^{ret}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$ whenever it is the case that $\mathsf{vis}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$.

Thus, even the final output $(\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}})$ will satisfy the containment.

We shall prove another interesting result pertaining to the conflict relations of two visibility relations extending the same reads-from relations, with one visibility relation contained inside another.

Lemma 24. *Let* rf_ℓ *be a reads-from relation over the well defined fragment* H_ℓ *and let* vis_ℓ *and* vis'_ℓ *be two visibility relations over* H_ℓ*, both extending* rf_ℓ*. Then,* $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell) \subseteq (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell) \cup (\mathsf{vis}'_\ell)_{\mathsf{Write}})^+$

Proof. Suppose $(o'', o') \in \mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}_\ell)$. That implies that there exists a read operation o such that $o'', o' \in \mathit{MaxRelWrites}_{\mathsf{vis}_\ell}(o)$ and $o' = \mathsf{rf}_\ell^{-1}(o)$.

Since $\mathsf{vis}_\ell \subseteq \mathsf{vis}'_\ell$, it implies that $o'', o' \in \mathit{RelWrites}_{\mathsf{vis}'_\ell}(o)$.

We consider two cases.

Suppose $o'' \in \mathit{MaxRelWrites}_{\mathsf{vis}'_\ell}(o)$, then by definition, $(o'', o') \in \mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell)$. Therefore, in this case $(o'', o') \in (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell) \cup (\mathsf{vis}'_\ell)_{\mathsf{Write}})^+$.

Suppose $o'' \notin \mathit{MaxRelWrites}_{\mathsf{vis}'_\ell}(o)$. Then, this implies that o'' is not a maximal write in the vis'_ℓ view of o restricted to its related writes. Thus, either $o'' \xrightarrow{(\mathsf{vis}'_\ell)_{\mathsf{Write}}} o'$ or there exists a path from $o'' \xrightarrow{(\mathsf{vis}'_\ell)_{\mathsf{Write}}} o_1 \xrightarrow{(\mathsf{vis}'_\ell)_{\mathsf{Write}}} \ldots \xrightarrow{(\mathsf{vis}'_\ell)_{\mathsf{Write}}} o_k \xrightarrow{(\mathsf{vis}'_\ell)_{\mathsf{Write}}} o'''$ where $o''' \in \mathit{MaxRelWrites}_{\mathsf{vis}'_\ell}(o)$ and each of $o_1, \ldots, o_k \in \mathit{RelWrites}_{\mathsf{vis}'_\ell}(o)$. In this case too, either $o''' = o'$ or $(o''', o') \in \mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell)$. Thus even in this case $(o'', o') \in (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell) \cup (\mathsf{vis}'_\ell)_{\mathsf{Write}})^+$.

With this we can now prove the correctness of Theorem 13. We need to prove the following:

For a hybrid read-write history $H = (\mathcal{O}, \mathsf{so})$, weak and strong consistency criteria α_w, α_s and multilevel constraints ϕ, the procedure TestMultiLevelCorrectness returns a witness $(\mathsf{rf}, \mathsf{vis}_{\mathsf{weak}}, \mathsf{vis}_{\mathsf{strong}})$ over H iff H is multi-level correct with respect to α_w, α_s and ϕ.

Proof. Suppose the hybrid history H is multi-level correct with respect to the consistency criteria $\alpha_w = \beta_w \wedge \gamma_w$, $\alpha_s = \beta_s \wedge \gamma_s$, and multilevel constraints ϕ. Then, by theorem 12, there exists a reads-from relation rf and visibility relations $\mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}'_{\mathsf{strong}}$ over H_{weak} and H_{strong} extending $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ respectively such that

- $H_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{weak}} \models \beta_w$
- $H_{\mathsf{strong}}, \mathsf{vis}'_{\mathsf{strong}} \models \beta_s$
- $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \phi$

Since the procedure, iterates through all possible Reads-From relation, if it returns before encountering the rf mentioned earlier, then we have nothing to prove. Suppose it does not return. Then, we will consider the iteration with the Reads-From relation being rf.

Note that since $\mathsf{vis}^{\min}_{\mathsf{weak}}$ and $\mathsf{vis}^{\min}_{\mathsf{strong}}$ are extensions of $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ via the procedure BuildMinVisSingle, by Lemma 19, we have $\mathsf{vis}^{\min}_{\mathsf{weak}} \subseteq \mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}^{\min}_{\mathsf{strong}} \subseteq \mathsf{vis}'_{\mathsf{strong}}$.

Now, suppose for $\mathsf{total}(\mathtt{vis})$ is a subformula in α_w. Then $\mathsf{vis}'_{\mathsf{weak}}$ is a total order. Similarly if $\mathsf{total}(\mathtt{vis})$ is a subformula in α_s, then $\mathsf{vis}'_{\mathsf{strong}}$ is a total order.

For $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$,since we iterate through all the total orders extending $\mathsf{vis}^{\min}_{\ell}$, if the procedure returns before the iteration reaches vis'_{ℓ}, then, there is nothing to prove. Suppose, the procedure returns with none of the prior total orders extending $\mathsf{vis}^{\min}_{\ell}$. Then we consider the case where the iterating variable vis_{ℓ} is the total order vis'_{ℓ}.

On the other hand, if $\mathsf{total}\mathtt{vis}$ is not a subformula in α_w or α_s, then we would set the corresponding vis_{ℓ} to $\mathsf{vis}^{\min}_{\ell}$. In both these cases, we can notice that $\mathsf{vis}_{\ell} \subseteq \mathsf{vis}'_{\ell}$.

Now, we obtain $(\mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}})$ by invoking ComputeStableExtension with $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$. By Lemma 22, $H_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}} \models \mathit{VisBasic}(\alpha_w)$ $H_{\mathsf{strong}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}} \models \mathit{VisBasic}(\alpha_s)$ and $H, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}} \models \phi$.

Further, by Lemma 23, for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$, $\mathsf{vis}^{\mathsf{stable}}_{\ell} \subseteq \mathsf{vis}'_{\ell}$. Which implies that if $\mathsf{total}(\mathtt{vis})$ is a subformula in the ℓ-consistency criteria then, $\mathsf{vis}^{\mathsf{stable}}_{\ell}$ is a total order as vis'_{ℓ} is.

From this, we can conclude that $H_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}} \models \beta_s$ in addition to $H, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}} \models \phi$.

Now we check $H, \mathsf{rf}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}}$ for bad patterns.

Note that, $(H, \mathsf{rf}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}})$ cannot have BADVISIBILITY, THINAIR, BADINITREAD or BADREAD bad patterns, since that would imply the existence of those bad patterns in $(H, \mathsf{rf}, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}')$ since $\mathsf{vis}^{\mathsf{stable}}_{\ell}$ is contained within vis'_{ℓ} for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$.

We will show by contradiction that BADARB bad pattern doesn't exist for $(H, \mathsf{rf}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{weak}}, \mathsf{vis}^{\mathsf{stable}}_{\mathsf{strong}})$ doesn't exist. Suppose this bad pattern did exist. Then, there is a cycle $C = o_1 \xrightarrow{\sigma_1} o_2 \xrightarrow{\sigma_2} \ldots \xrightarrow{\sigma_n} o_1$ where each σ_i is one of $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}^{\mathsf{stable}}_\ell)$ or $(\mathsf{vis}^{\mathsf{stable}}_\ell)_{\mathsf{Write}}$ for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$

Note that since $\mathsf{vis}^{\mathsf{stable}}_\ell \subseteq \mathsf{vis}'_\ell$, in the Cycle C above, we can rewrite the edge $o_i \xrightarrow{\mathsf{vis}^{\mathsf{stable}}_\ell} o_{i+1}$ by $o_i \xrightarrow{\mathsf{vis}'_\ell} o_{i+1}$.

Further from Lemma 24, we have $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}^{\mathsf{stable}}_\ell) \subseteq (\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell) \cup (\mathsf{vis}'_\ell)_{\mathsf{Write}})^+$. Which means that the any edge $o_i \xrightarrow{\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}^{\mathsf{stable}}_\ell)} o_{i+1}$ in the cycle C can be replaced by a path $o_i \xrightarrow{\sigma'_1} \ldots \xrightarrow{\sigma'_{n'}} o_{i+1}$ where each σ'_k is either $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell)$ or $(\mathsf{vis}'_\ell)_{\mathsf{Write}}$. Thus, we get a cycle C' from C whose edges comprise of $\mathsf{CF}(\mathsf{rf}_\ell, \mathsf{vis}'_\ell)$ and $(\mathsf{vis}'_\ell)_{\mathsf{Write}}$ for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$. Thus, BADARB bad pattern exists for $(H, \mathsf{rf}, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}})$, which is a contradiction. Thus, if H is correct, then we have proved that the procedure TestMultiLevelCorrectness produces a satisfying witness.

Conversely we will show that if TestMultiLevelCorrectness produces a satisfying witness then the hybrid history H is multi-level correct.

Suppose rf is the witness reads-from relation and $\mathsf{vis}_{\mathsf{weak}}$ and $\mathsf{vis}_{\mathsf{strong}}$ are the visibility relations extending rf which are extended via ComputeStableExtension to obtain $\mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}'_{\mathsf{strong}}$. Suppose that none of the bad patterns exist for $(H, \mathsf{rf}, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}})$.

By lemma22, we know that

- $H_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{weak}} \models \mathit{VisBasic}(\alpha_w)$
- $H_{\mathsf{strong}}, \mathsf{vis}'_{\mathsf{strong}} \models \mathit{VisBasic}(\alpha_s)$
- $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \phi$.

If for $\ell \in \{\mathsf{weak}, \mathsf{strong}\}$ if the corresponding consistency criteria contains the subformula $\mathsf{total}(\mathtt{vis})$. Then the iterating variable vis_ℓ would have been a total order. By lemma 22, we know that $\mathsf{vis}_\ell \subseteq \mathsf{vis}'_\ell$. Suppose $\mathsf{vis}_\ell \subsetneq \mathsf{vis}'_\ell$, it implies that vis'_ℓ has atleast one additional edges between the operations of $\mathcal{O}_\ell$ over what is present in vis_ℓ. However, since vis_ℓ is a total order, it implies that in the additional edges introduce a cycle in vis'_ℓ. But this is not the case since BADVISIBILITY bad-pattern would have caught it. Hence $\mathsf{vis}'_\ell = \mathsf{vis}_\ell$ in this case which implies that if total is a subformula in the consistency criteria for level ℓ, then, $H_\ell, \mathsf{vis}'_\ell \models \mathsf{total}(\mathtt{vis})$.

Thus, we can conclude that there exists a reads-from relation rf and weak and strong visibility relations $\mathsf{vis}'_{\mathsf{weak}}$ and $\mathsf{vis}'_{\mathsf{strong}}$ extending $\mathsf{rf}_{\mathsf{weak}}$ and $\mathsf{rf}_{\mathsf{strong}}$ respectively such that $H_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{weak}} \models \beta_w$, $H_{\mathsf{strong}}, \mathsf{vis}'_{\mathsf{strong}} \models \beta_s$, $H, \mathsf{vis}'_{\mathsf{weak}}, \mathsf{vis}'_{\mathsf{strong}} \models \phi$, and none of the bad patterns exist.By theorem 12, this implies that the hybrid history H is multi-level correct with respect to α_w, α_s, ϕ