Algebraic Laws for Weak Consistency

Andrea Cerone[1], Alexey Gotsman[2], and Hongseok Yang[3]

1 **Imperial College London, UK,** acerone@ic.ac.uk
2 **IMDEA Software Institute, Madrid, Spain,** alexey.gotsman@imdea.org
3 **University of Oxford, UK,** hongseok.yang@cs.ox.ac.uk

Abstract

Modern distributed systems often rely on so called weakly-consistent databases, which achieve scalability by sacrificing the consistency guarantee of distributed transaction processing. Such databases have been formalised in two different styles, one based on abstract executions and the other based on dependency graphs. The choice between these styles has been made according to intended applications. The former has been used for specifying and verifying the implementation of these databases, while the latter for proving properties of client programs of the databases. In this paper, we present a set of novel algebraic laws (i.e. inequations) that connect these two styles of specifications. The laws relate binary relations used in a specification based on abstract executions, to those used in a specification based on dependency graphs. We then show that this algebraic connection gives rise to so called robustness criteria, conditions which ensure that a client program of a weakly-consistent database does not exhibit anomalous behaviours due to weak consistency. These criteria make it easy to reason about these client programs, and may become a basis for dynamic or static program analyses. For a certain class of consistency models specifications, we prove a full abstraction result that connects the two styles of specifications.

1 Introduction

Modern distributed systems often rely on databases that achieve scalability by sacrificing the consistency guarantee of distributed transaction processing. These databases are said to implement weak consistency models. Such weakly-consistent databases allow for faster transaction processing, but exhibit anomalous behaviours, which do not arise under a database with a strong consistency guarantee, such as serialisability. Two important problems for the weakly-consistent databases are: (i) to find elegant formal specifications of their consistency models and to prove that these specifications are correctly implemented by protocols used in the databases; (ii) to develop effective reasoning techniques for applications running on top of such databases. These problems have been tackled by using two different formalisms, which model the run-time behaviours of weakly-consistent databases differently.

When the goal is to verify the correctness of a protocol implementing a weak consistency model, the run-time behaviour of a distributed database is often described in terms of *abstract executions* [11], which abstract away low-level implementation details of the database (§2). An example of abstract execution is depicted in Figure 1; ignore the bold edges for the moment. It comprises four transactions, T_0, T_1, T_2, and S; transaction T_0 initializes the value of an object acct to 0; transactions T_1 and T_2 update the value of acct to 50 and 25, respectively, after reading its initial value; transaction S reads the value of acct. In this abstract execution, both the updates of T_1 and T_2 are **VIS**ible to transaction S, as witnessed by the two VIS-labelled edges: $T_1 \xrightarrow{\mathsf{VIS}} S$ and $T_2 \xrightarrow{\mathsf{VIS}} S$. On the other hand, the

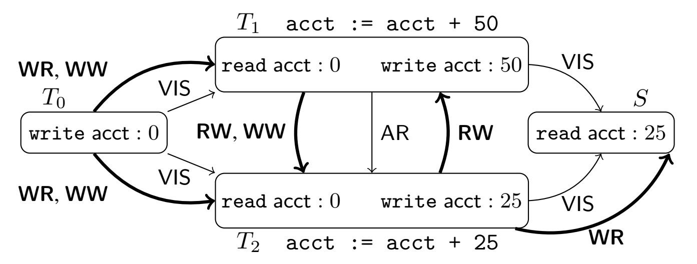


Figure 1 An example of abstract execution and of dependency graph.

Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

update of T_1 is not visible to T_2, and vice versa, as indicated by the absence of an edge labelled with VIS between these transactions. Intuitively, the absence of such an edge means that T_1 and T_2 are executed concurrently. Because S sees T_1 and T_2, as indicated by VIS-labelled edges from T_1 and T_2 to S, the result of reading the value of acct in S must be one of the values written by T_1 and T_2. However, because these transactions are concurrent, there is a race, or *conflict*, between them. The AR-labelled edge connecting T_1 to T_2, is used to **AR**bitrate the conflict: it states that the update of T_1 is older than the one of T_2, hence the query of acct in S returns the value written by the latter.

The style of specifications of consistency models in terms of abstract executions can be given by imposing constraints over the relations VIS, AR (§2.1). A set of transactions $\mathcal{T} = \{T_1, T_2, \cdots\}$, called *history*, is allowed by a consistency model specification if it is possible to exhibit two witness relations VIS, AR over $\mathcal{T}$ such that the resulting abstract execution satisfies the constraints imposed by the specification. For example, *serialisability* can be specified by requiring that the relation VIS should be a strict total order. The set of transactions $\{T_0, T_1, T_2, S\}$ from Figure 1 is not serialisable: it is not possible to choose a relation VIS such that the resulting abstract execution relates the transactions T_1, T_2 and the results of the reads are consistent with visible updates.

Specifications of consistency models using abstract executions have been used in the work on proving the correctness of protocols implementing weak consistency models, as well as on justifying operational, implementation-dependent descriptions of these models [9, 10, 11, 13].

The second formalism used to define weak consistency models is based on the notion of *dependency graphs* [2], and it has been used for proving properties of client programs running on top of a weakly-consistent database. Dependency graphs capture the data dependencies of transactions at run-time (§3); the transactions $\{T_0, T_1, T_2, S\}$ depicted above, together with the bold edges but without normal edges, constitute an example of dependency graph. The edge $T_2 \xrightarrow{\mathsf{WR(acct)}} S$[1] means that the read of acct in transaction S returns the value written by transaction T_2, and the edges $T_0 \xrightarrow{\mathsf{WR(acct)}} T_1$ and $T_0 \xrightarrow{\mathsf{WR(acct)}} T_2$ mean something similar. The edge $T_1 \xrightarrow{\mathsf{WW(acct)}} T_2$ denotes a *write-write dependency*, and says that the write to acct in T_2 supersedes the write to the same object in T_1. The remaining edges $T_1 \xrightarrow{\mathsf{RW(acct)}} T_2$ and $T_2 \xrightarrow{\mathsf{RW(acct)}} T_1$ express *read-write anti-dependencies*. The former means that T_1 reads a value for object acct which is older than the value written by T_2.

When using dependency graphs, consistency models are specified as sets of transactions (or histories) for which there exist WR, WW, RW relations that satisfy certain properties, usually stated as particular relations being acyclic [6, 14]. Because dependencies of transactions can be over-approximated at the compilation time, specifications of consistency models in terms of dependency graphs have been widely used for manually or automatically reasoning about properties of client programs of weakly-consistent databases [2, 16, 25]. They have also been used in the complexity and undecidability results for verifying implementations of consistency models [7].

Our ultimate aim is to reveal a deep connection between these two styles of specifying weak consistency models, which was hinted at for specific consistent models in the literature. Such a connection would, for instance, give us a systematic way to derive a specification of a weak consistency model based on dependency graphs from the specification based on abstract executions, while ensuring that the original and the derived specifications are equivalent in a sense. In doing so, it would enable us to prove properties about client programs of a weakly-consistent database using techniques based on dependency graphs [7, 14, 15] even when the consistency model of the database is specified in terms of abstract executions.

In this paper, we present our first step towards this ultimate aim. We describe a novel *algebraic* connection between the two styles of specifications for weak consistency models. We present algebraic

[1] For simplicity, references to the object acct have been removed from the dependencies of Figure 1.

laws (i.e. inequations) that connect the $\mathsf{VIS}, \mathsf{AR}$ relations for abstract executions, to dependencies and anti-dependencies for dependency graphs (§4). For several consistency models, these laws give rise to so called robustness criteria for client programs, conditions ensuring that a program only exhibits serialisable behaviours even when it runs under a weak consistency model [6, 8, 16]. These criteria are derived as follows: using our algebraic laws, we derive for a given consistency model, a property of the form $R \cap \mathsf{Id} \subseteq \emptyset$, where R is an expression from the Kleene Algebra with Tests [20] whose ground terms are run-time dependencies of transactions, and tests are properties over transactions. These properties in turn give a necessary condition for the presence of cycles in dependency graphs in the model. We can then check for the absence of such cycles in an application at compile time: because dependency graphs of serialisable databases are always acyclic, this ensures that said application only exhibits serialisable behaviours.

As another contribution, we devise a proof technique for inferring, given a consistency model specified using abstract executions, an equivalent specification in terms of dependency graphs (§5). This proof technique is sound for a restricted, yet meaningful class of consistency model specifications, which we call simple. The technique works as follows. For each simple specification of consistency model, we define a system of inequations over run-time dependencies of transactions; there the unknowns embed a relation R whose acyclicity represents a necessary and sufficient condition of dependency graphs allowed by the consistency model. By solving said system of inequations, one immediately obtains an equivalent specification of the consistency model using dependency graphs.

One key insight in our algebraic laws is that there is a correspondence between the AR relation and a novel relation that we call *anti-visibility*, which encompasses anti-dependencies. The exact nature of this relation depends from the specification of a particular consistency model. To keep the discussion simple, we adopt causal consistency [22] as the weakest consistency model; we discuss a possible generalisation to a wider class of consistency models in Appendix (§B).

2 Abstract Executions

We consider a database storing objects in $\mathsf{Obj} = \{x, y, \cdots\}$, which for simplicity we assume to be integer-valued. Client programs can interact with the database by executing operations from a set Op, grouped inside *transactions*. We leave the set Op unspecified, apart from requiring that it contains read and write operations over objects: $\{\mathtt{write}(x, n), \mathtt{read}(x, n) \mid x \in \mathsf{Obj}, n \in \mathbb{N}\} \subseteq \mathsf{Op}$.

Histories. To specify a consistency model, we first define the set of all client-database interactions allowed by the model. We start by introducing (run-time) *transactions* and *histories*, which record such interactions in a single computation. Transactions are elements from a set $\mathbb{T} = \{T, S, \cdots\}$; the operations executed by transactions are given by a function $\mathsf{behav} : \mathbb{T} \to 2^{\mathsf{Op}}$, which maps a transaction T to a set of operations that are performed by the transaction and can be observed by other transactions. We often abuse notations and just write $o \in T$ (or $T \ni o$) instead of $o \in \mathsf{behav}(T)$. We adopt similar conventions for $\mathcal{O} \subseteq \mathsf{behav}(T)$ and $\mathcal{O} = \mathsf{behav}(T)$ where $\mathcal{O}$ is a subset of operations.

We assume that transactions enjoy *atomic visibility* : for each object x, (i) a transaction S never observes two different writes to x from a single transaction T and (ii) it never reads two different values of x. Formally, the requirements are that if $T \ni (\mathtt{write}\ x : n)$ and $T \ni (\mathtt{write}\ x : m)$, or $T \ni (\mathtt{read}\ x : n)$ and $T \ni (\mathtt{read}\ x : m)$, then $n = m$. Our treatment of atomic visibility is taken from our previous work on transactional consistency models [13]. Atomic visibility is guaranteed by many consistency models [4, 19, 26]. We point out that although we focus on transactions in distributed systems in the paper, our results apply to weak shared-memory models [3]; there a transaction T is the singleton set of a read operation ($T = \{\mathtt{read}\ x : n\}$), that of a write operation ($T = \{\mathtt{write}\ x : n\}$), or the set of read and write representing a *compare and set* operation ($T = \{\mathtt{read}\ x : n,\ \mathtt{write}\ x : m\}$).

For each object x, we let $\mathsf{Writes}_x := \{T \mid \exists n.\ (\mathtt{write}\ x : n) \in T\}$ and $\mathsf{Reads}_x := \{T \mid \exists n,\ (\mathtt{read}\ x : n) \in \mathcal{T}\}$ be the sets of transactions that write to and read from x, respectively.

▸ **Definition 1.** A *history* $\mathcal{T}$ is a set of transactions $\{T_1, T_2, \cdots, T_n\}$.

Consistency Models. A consistency model Γ is a set of histories that may arise when client programs interact with the database. To define Γ formally, we use the notion of abstract executions: these are histories augmented with two relations, called *visibility* and *arbitration*.

▸ **Definition 2.** An *abstract execution* $\mathcal{X}$ is a tuple $(\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ consisting of a history $\mathcal{T}$ and relations $\mathsf{VIS}, \mathsf{AR} \subseteq (\mathcal{T} \times \mathcal{T})$ on transactions such that $\mathsf{VIS} \subseteq \mathsf{AR}$ and AR is a strict total order[2].

We often write $T \xrightarrow{\mathsf{VIS}} S$ for $(T, S) \in \mathsf{VIS}$, and similarly for other relations. For each abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$, we let $\mathcal{T}_{\mathcal{X}} := \mathcal{T}$, $\mathsf{VIS}_{\mathcal{X}} := \mathsf{VIS}$, and $\mathsf{AR}_{\mathcal{X}} := \mathsf{AR}$. We denote the set of abstract executions by $\mathsf{Executions}$.

In an abstract execution $\mathcal{X}$, $T \xrightarrow{\mathsf{VIS}_{\mathcal{X}}} S$ means that the read operations in S may depend on the updates of T, while $T \xrightarrow{\mathsf{AR}_{\mathcal{X}}} S$ means that the update operations of S supersede those performed by T. Naturally, one would expect that the value fetched by read operations in a transaction T is the most up-to-date one among all the values written by transactions visible to T. For simplicity, we assume that such a transaction always exists.

▸ **Definition 3.** An abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ respects the *Last Write Win* (**LWW**) policy, if for all $T \in \mathcal{T}$ and all $(\mathtt{read}\ x : n) \in T$, the set $\mathcal{T}' := \left(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x\right)$ is not empty, and $\max_{\mathsf{AR}}(\mathcal{T}') \ni \mathtt{write}\ x : n$ where $\max_{\mathsf{AR}}(\mathcal{T}')$ is the AR-supremum of $\mathcal{T}'$.

▸ **Definition 4.** An abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ *respects causality* if VIS is transitive. Any abstract execution that respects both causality and the LWW policy is said to be *valid*.

We always assume an abstract execution to be valid, unless otherwise stated. Causality is respected by all abstract executions allowed by several interesting consistency models. They also simplify the mathematical development of our results. In (§B), we explain how our results can be generalised for consistency models that do not respect causality. We also discuss how the model can be generalised to account for sessions and session guarantees [27].

We can specify a consistency model using abstract executions in two steps. First, we identify properties on abstract executions, or *axioms*, that formally express an informal consistency guarantee, and form a set with the abstract executions satisfying the properties. Next, we project abstract executions in this set to underlying histories, and define a consistency model Γ to be the set of resulting histories.

Abstract executions hide low-level operational details of the interaction between client programs and weakly-consistent databases. This benefit has been exploited for proving that such databases implement intended consistency models [9, 10, 11, 13, 17].

2.1 Specification of Weak Consistency Models

In this section we introduce a simple framework for specifying consistency models using the style of specification discussed above. In our framework, axioms of consistency models relate the visibility and arbitration relations via inequations of the form $R_1 \ ; \mathsf{AR}_{\mathcal{X}} \ ; R_2 \subseteq \mathsf{VIS}_{\mathcal{X}}$, where R_1 and R_2 are particular relations over transactions, and $\mathcal{X}$ is an abstract executions. As we will explain later, axioms of this form establish a necessary condition for two transactions in an abstract execution $\mathcal{X}$ to be related by $\mathsf{VIS}_{\mathcal{X}}$, i.e. they cannot be executed concurrently. Despite its simplicity, the framework is

[2] A relation $R \subseteq \mathcal{T} \times \mathcal{T}$ is a strict (partial) order if it is transitive and irreflexive; it is total if for any $T, S \in \mathcal{T}$, either $T = S$, $(T, S) \in R$ or $(S, T) \in R$.

expressive enough to capture several consistency models for distributed databases [13, 21]; as we will show in §4, one of the benefits of this simplicity is that we can infer robustness criteria of consistency models in a systematic way.

As we will see, the relations R_1, R_2 in axioms of the form above, may depend on the visibility relation of the abstract execution $\mathcal{X}$. To define such relations, we introduce the notion of *specification function*.

▸ **Definition 5.** A function $\rho : 2^{(\mathbb{T} \times \mathbb{T})} \rightarrow 2^{(\mathbb{T} \times \mathbb{T})}$ is a *specification function* if for every relation $R \subseteq \mathcal{T} \times \mathcal{T}$, we have that $\rho(R) = \rho(\mathcal{T} \times \mathcal{T}) \cap R?$, where $R?$ is the reflexive closure of R. A *consistency guarantee* or simply *guarantee* is a pair of specification functions (ρ, π).

Definition 5 ensures that specification functions are defined locally: for any $R_1, R_2 \in \mathcal{T} \times \mathcal{T}$, $\rho(R_1 \cup R_2) = \rho(R_1) \cup \rho(R_2)$, and in particular for any $R \subseteq \mathcal{T} \times \mathcal{T}$, $\rho(R) = \left(\bigcup_{T,S \in \mathcal{T}} \rho(\{(T,S)\})\right) \cap R?$. The reflexive closure in Definition 5 is needed because we will always apply specification functions to irreflexive relations (namely, the visibility relation of abstract executions), although the result of this application need not be irreflexive. For example, $\rho_{\mathsf{Id}}(R) := \mathsf{Id}$ is a valid specification function.

Each consistency guarantee (ρ, π) defines, for each abstract execution $\mathcal{X}$, an inequation of the form $\rho(\mathsf{VIS}_{\mathcal{X}}) \ ; \ \mathsf{AR}_{\mathcal{X}} \ ; \ \pi(\mathsf{VIS}_{\mathcal{X}}) \subseteq \mathsf{VIS}_{\mathcal{X}}$: if this inequation is satisfied by $\mathcal{X}$, we say that $\mathcal{X}$ satisfies the consistency guarantee (ρ, π). Consistency guarantees impose a condition on when two transactions T, S in an abstract execution $\mathcal{X}$ are not allowed to execute concurrently, i.e. they must be related by a $\mathsf{VIS}_{\mathcal{X}}$ edge. By definition of abstract executions, visibility edges in abstract execution cannot contradict arbitration edges, hence it is only natural that the order in which the transactions T, S above are executed is determined by the arbitration order: in fact, the definition of specification function ensures that $\rho(\mathsf{VIS}_{\mathcal{X}}) \subseteq \mathsf{VIS}_{\mathcal{X}}?$ and $\pi(\mathsf{VIS}_{\mathcal{X}}) \subseteq \mathsf{VIS}_{\mathcal{X}}?$, so that $(\rho(\mathsf{VIS}_{\mathcal{X}}) \ ; \ \mathsf{AR}_{\mathcal{X}} \ ; \ \pi(\mathsf{VIS}_{\mathcal{X}})) \subseteq \mathsf{AR}_{\mathcal{X}}$ for all abstract executions $\mathcal{X}$.

▸ **Definition 6.** A *consistency model specification* Σ or *x-specification* is a set of consistency guarantees $\{(\rho_i, \pi_i)\}_{i \in I}$ for some index set I.

We define $\mathsf{Executions}(\Sigma)$ to be the set of abstract executions that satisfy all the consistency guarantees of Σ. We let $\mathsf{modelOf}(\Sigma) := \{\mathcal{T}_{\mathcal{X}} \mid \mathcal{X} \in \mathsf{Executions}(\Sigma)\}$.

Examples of Consistency Model Specifications. Figure 2 shows several examples of specification functions and consistency guarantees. In the figure, Id is the identity relation over transactions, and we use the relations $[\mathcal{T}] := \{(T,T) \mid T \in \mathcal{T}\}$ and $[o] := \{(T,T) \mid T \ni o\}$ for $\mathcal{T} \subseteq \mathbb{T}$ and $o \in \mathsf{Op}$. The guarantees in the figure can be composed together to specify several consistency models: we give some examples of them below. Each of these consistency models allows different kinds of anomalies: due to lack of space, these are illustrated in (§A).

Causal Consistency: This is the weakest consistency model in the paper. It is specified by $\Sigma_{\mathsf{CC}} = \emptyset$. In this case, all abstract executions in $\mathsf{Executions}(\Sigma_{\mathsf{CC}})$ respect causality. The execution in Figure 1 is an example in $\mathsf{Executions}(\Sigma_{\mathsf{CC}})$. Causal consistency has been implemented for geo-replicated databases [22]. Our specification coincides with the one given in [13].

Function		Definition
$\rho_{\mathsf{Id}}(R)$	=	Id
$\rho_{\mathsf{SI}}(R)$	=	$R \setminus \mathsf{Id}$
$\rho_x(R)$	=	$[\mathsf{Writes}_x]$
$\rho_S(R)$	=	$[\mathtt{SerTx}]$

Guarantee	Associated Axiom
$(\rho_{\mathsf{Id}}, \rho_{\mathsf{Id}})$	$\mathsf{AR} \subseteq \mathsf{VIS}$
$(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$	$\mathsf{AR} \ ; \ \mathsf{VIS} \subseteq \mathsf{VIS}$
(ρ_x, ρ_x)	$[\mathsf{Writes}_x] \ ; \ \mathsf{AR} \ ; \ [\mathsf{Writes}_x] \subseteq \mathsf{VIS}$
(ρ_S, ρ_S)	$[\mathtt{SerTx}] \ ; \ \mathsf{AR} \ ; \ [\mathtt{SerTx}] \subseteq \mathsf{VIS}$

Figure 2 Some Specification Functions and Consistency Guarantees

Red-Blue Consistency: This model extends causal consistency by marking a subset of transactions as serialisable, and ensuring that no two such transactions appear to execute concurrently. It is implemented in [21]. We model red-blue consistency via the x-specification $\Sigma_{\mathsf{RB}} = \{(\rho_S, \rho_S)\}$. In the definition of ρ_S, an element $\mathtt{SerTx} \in \mathsf{Op}$ is used to mark transactions as serialisable, and the specification requires that in every execution $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{RB}})$, any two transactions $T, S \ni \texttt{SerTx}$ in $\mathcal{X}$ be compared by $\mathsf{VIS}_{\mathcal{X}}$. The abstract execution from Figure 1 is included in $\mathsf{Executions}(\Sigma_{\mathsf{RB}})$, but if it were modified so that transactions T_1, T_2 were marked as serialisable, then the result would not belong to $\mathsf{Executions}(\Sigma_{\mathsf{RB}})$.

Parallel Snapshot Isolation (PSI): This model strengthens causal consistency by enforcing the *Write Conflict Detection* property: transactions writing to one same object do not execute concurrently. PSI has been implemented in [24, 26]. We let $\Sigma_{\mathsf{PSI}} = \{(\rho_x, \rho_x)\}_{x \in \mathsf{Obj}}$: then every execution $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$ satisfies the inequation $([\mathsf{Writes}_x]\ ;\ \mathsf{AR}_{\mathcal{X}}\ ;\ [\mathsf{Writes}_x]) \subseteq \mathsf{VIS}_{\mathcal{X}}$, for all $x \in \mathsf{Obj}$. This specification is equivalent to the one presented in [13].

Snapshot Isolation (SI): This consistency model strengthens PSI by requiring that, in executions, the set of transactions visible to any transaction T is a prefix of the arbitration relation. Formally, we let $\Sigma_{\mathsf{SI}} = \Sigma_{\mathsf{PSI}} \cup \{(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})\}$. The consistency guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$ ensures that any abstract execution $\mathcal{X} \in \mathsf{Executions}(\mathsf{SI})$ satisfies the property $(\mathsf{AR}_{\mathcal{X}}\ ;\ \mathsf{VIS}_{\mathcal{X}}) \subseteq \mathsf{VIS}_{\mathcal{X}}$[3]. In [13], we have proved that this specification is equivalent to the original, operational one [5].

Similarly to what we did to specify Red-Blue consistency, we can strengthen SI by allowing the possibility to mark transactions as serialisable. The resulting x-specification is $\Sigma_{\mathsf{SI+SER}} = \Sigma_{\mathsf{SI}} \cup \{(\rho_S, \rho_S)\}$. This x-specification captures a fragment of Microsoft SQL server, which allows the user to select the consistency model at which a transaction should run [1].

Serialisability: Executions in this consistency model require the visibility relation to be total. This can be formalised via the x-specification $\Sigma_{\mathsf{SER}} := \{(\rho_{\mathsf{Id}}, \rho_{\mathsf{Id}})\}$. Any $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SER}})$ is such that $\mathsf{AR}_{\mathcal{X}} \subseteq \mathsf{VIS}_{\mathcal{X}}$, thus enforcing $\mathsf{VIS}_{\mathcal{X}}$ to be a strict total order.

3 Dependency Graphs

We present another style of specification for consistency models based on dependency graphs, introduced in [2]. These are structures that capture the data-dependencies between transactions accessing one same object. Such dependencies can be over approximated at compilation time. For this reason, they have found use in static analysis [6, 14, 15, 16] for programs running under a weak consistency model.

▸ **Definition 7.** A ***dependency graph*** is a tuple $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$, where $\mathcal{T}$ is a history and

1. $\mathsf{WR} : \mathsf{Obj} \to 2^{\mathcal{T} \times \mathcal{T}}$ is such that:
   (a) $\forall T, S \in \mathcal{T}. \forall x. T \xrightarrow{\mathsf{WR}(x)} S \implies T \neq S \wedge \exists n. (T \ni \texttt{write}\ x : n) \wedge (S \ni \texttt{read}\ x : n)$,
   (b) $\forall S \in \mathcal{T}. \forall x. (S \ni \texttt{read}\ x : n) \implies \exists T. T \xrightarrow{\mathsf{WR}(x)} S$,
   (c) $\forall T, T', S \in \mathcal{T}. \forall x. (T \xrightarrow{\mathsf{WR}(x)} S \wedge T' \xrightarrow{\mathsf{WR}(x)} S) \implies T = T'$;
2. $\mathsf{WW} : \mathsf{Obj} \to 2^{\mathcal{T} \times \mathcal{T}}$ is such that for every $x \in \mathsf{Obj}$, $\mathsf{WW}(x)$ is a strict, total order over Writes_x;
3. $\mathsf{RW} : \mathsf{Obj} \to 2^{\mathcal{T} \times \mathcal{T}}$ is such that $S \xrightarrow{\mathsf{RW}(x)} T$ iff $S \neq T$ and $\exists T'. T' \xrightarrow{\mathsf{WR}(x)} S \wedge T' \xrightarrow{\mathsf{WW}(x)} T$.

Given a dependency graph $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$, we let $\mathcal{T}_{\mathcal{G}} := \mathcal{T}$, $\mathsf{WR}_{\mathcal{G}} := \mathsf{WR}$, $\mathsf{WW}_{\mathcal{G}} := \mathsf{WW}$, $\mathsf{RW}_{\mathcal{G}} := \mathsf{RW}$. The set of all dependency graphs is denoted as Graphs. Sometimes, we commit an abuse of notation and use the symbol WR to denote the relation $\bigcup_{x \in \mathsf{Obj}} \mathsf{WR}(x)$, and similarly for WW and RW. The actual meaning of WR will always be clear from the context.

[3] To be precise, the property induced by the guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$ is $(\mathsf{AR}_{\mathcal{X}}\ ;\ (\mathsf{VIS}_{\mathcal{X}} \backslash \mathsf{Id})) \subseteq \mathsf{AR}_{\mathcal{X}}$. However, since $\mathsf{VIS}_{\mathcal{X}}$ is an irreflexive relation, $\mathsf{VIS}_{\mathcal{X}} \backslash \mathsf{Id} = \mathsf{VIS}_{\mathcal{X}}$. Also, note that $\rho(\mathcal{T}, R) = R$ is not a specification function, so we cannot replace the guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$ with $(\rho_{\mathsf{Id}}, \rho)$.

Let $\mathcal{G} \in \mathsf{Graphs}$. $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$ means that S reads the value of object x that has been written by T. By Definition 7, for any transaction $S \in \mathsf{Reads}_x$ there exists exactly one transaction T such that $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$. The relation $\mathsf{WW}_{\mathcal{G}}(x)$ establishes a total order in which updates over object x are executed by transactions. The relation $\mathsf{RW}_{\mathcal{G}}(x)$ takes the name of *anti-dependency*. $T \xrightarrow{\mathsf{RW}_{\mathcal{G}}(x)} S$ means that transaction T fetches some value for object x, but this is later updated by S. Given an abstract execution $\mathcal{X}$, we can extract a dependency graph $\mathsf{graph}(\mathcal{X})$ such that $\mathcal{T}_{\mathsf{graph}(\mathcal{X})} = \mathcal{T}_{\mathcal{X}}$.

▸ **Definition 8.** Let $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ be an execution. For $x \in \mathsf{Obj}$, we define $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}_{\mathcal{X}}, \mathsf{WW}_{\mathcal{X}}, \mathsf{RW}_{\mathcal{X}})$, where:

1. $T \xrightarrow{\mathsf{WR}_{\mathcal{X}}(x)} S \iff (S \ni \mathtt{read}\ x : \_) \wedge T = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(S) \cap \mathsf{Writes}_x)$;
2. $T \xrightarrow{\mathsf{WW}_{\mathcal{X}}(x)} S \iff T \xrightarrow{\mathsf{AR}} S \wedge T, S \in \mathsf{Writes}_x$;
3. $T \xrightarrow{\mathsf{RW}_{\mathcal{X}}(x)} S \iff S \neq T \wedge (\exists T'.T' \xrightarrow{\mathsf{WR}_{\mathcal{X}}(x)} T \wedge T' \xrightarrow{\mathsf{WW}_{\mathcal{X}}(x)} S))$.

▸ Proposition 9. For any valid abstract execution $\mathcal{X}$, $\mathsf{graph}(\mathcal{X})$ is a dependency graph.

Specification of Consistency Models using Dependency Graphs. We interpret a dependency graph $\mathcal{G}$ as a labelled graph whose vertices are transactions in $\mathcal{T}_x$, and whose edges are pairs of the form $T \xrightarrow{R} S$, where $R \in \{\mathsf{WR}_{\mathcal{G}}(x), \mathsf{WW}_{\mathcal{G}}(x)_{\mathcal{G}}, \mathsf{RW}_{\mathcal{G}}(x) \mid x \in \mathsf{Obj}\}$. To specify a consistency model, we define a style of specifications of consistency models in two steps. We first identify one or more conditions to be satisfied by dependency graphs. Such conditions require cycles of a certain form not to appear in a dependency graph. Then we define a consistency model by projecting the set of dependency graphs satisfying the imposed conditions into the underlying histories. This style of specification is reminiscent of the one used in the CAT [3] language for formalising weak memory models. In the following we treat the relations $\mathsf{WR}_{\mathcal{G}}(x), \mathsf{WW}_{\mathcal{G}}(x), \mathsf{RW}_{\mathcal{G}}(x)$ both as set-theoretic relations, and as edges of a labelled graph.

▸ **Definition 10.** A *dependency graph based specification*, or simply g-specification, is a set $\Delta = \{\delta_1, \cdots, \delta_n\}$, where for each $i \in \{1, \cdots, n\}$, δ_i is a function of type $\mathsf{Graphs} \to 2^{(\mathbb{T} \times \mathbb{T})}$ and satisfies $\delta_i(\mathcal{G}) \subseteq (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}} \cup \mathsf{RW}_{\mathcal{G}})^*$ for every $\mathcal{G} \in \mathsf{Graphs}$.

Given a g-specification Δ, we define $\mathsf{Graphs}(\Delta) = \{\mathcal{G} \in \mathsf{Graphs} \mid \forall \delta \in \Delta.\, \delta(\mathcal{G}) \cap \mathsf{Id} = \varnothing\}$, and we let $\mathsf{modelOf}(\Delta) = \{\mathcal{T} \mid \exists \mathsf{WR}, \mathsf{WW}, \mathsf{RW}.\, (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW}) \in \mathsf{Graphs}(\Delta)\}$.

The requirement imposed over the functions $\delta_1, \cdots, \delta_n$ ensures that, whenever $(T, S) \in \delta_i(\mathcal{G})$, for some dependency graph $\mathcal{G}$, then there exists a path in $\mathcal{G}$, that connects T to S. For $\Delta = \{\delta_i\}_{i=1}^n$ and $\mathcal{G} \in \mathsf{Graphs}$, the requirement that $\delta_i(\mathcal{G}) \cap \mathsf{Id} = \varnothing$ means that $\mathcal{G}$ does not contain any cycle $T_0 \xrightarrow{R_0} T_1 \xrightarrow{R_1} \cdots \xrightarrow{R_{n-1}} T_n$, such that $T_0 = T_n$, and $(R_0 \,;\, \cdots \,;\, R_{n-1}) \subseteq \delta_i(\mathcal{G})$.

Examples of g-specifications of consistency models. Below we give some examples of g-specifications for the consistency models presented in §2.

▸ **Theorem 11.**

1. *An execution $\mathcal{X}$ is serialisable iff* $\mathsf{graph}(\mathcal{X})$ *does not contain any cycle. That is,* $\mathsf{modelOf}(\Sigma_{\mathsf{SER}}) = \mathsf{modelOf}(\{\delta_{\mathsf{SER}}\})$*, where* $\delta_{\mathsf{SER}}(\mathcal{G}) = (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}} \cup \mathsf{RW}_{\mathcal{G}})^+$.
2. *An execution $\mathcal{X}$ is allowed by parallel snapshot isolation iff* $\mathsf{graph}(\mathcal{X})$ *has no cycle where all anti-dependency edges are over the same object. Let* $\delta_{\mathsf{PSI}_0}(\mathcal{G}) = (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^+$, $\delta_{\mathsf{PSI}(x)}(\mathcal{G}) = (\bigcup_{x \in \mathsf{Obj}}(\mathsf{WR}_{\mathcal{G}}(x) \cup \mathsf{WW}_{\mathcal{G}}(x))^* \,;\, \mathsf{RW}_{\mathcal{G}}(x))^+$*, and define* $\Delta_{\mathsf{PSI}} = \{\delta_{\mathsf{PSI}_0}\} \cup \{\delta_{\mathsf{PSI}(x)} \mid x \in \mathsf{Obj}\}$*. Then,* $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) = \mathsf{modelOf}(\Delta_{\mathsf{PSI}})$.
3. *An execution $\mathcal{X}$ is allowed by snapshot isolation iff* $\mathsf{graph}(\mathcal{X})$ *only admits cycles with at least two consecutive anti-dependency edge. That is,* $\mathsf{modelOf}(\Sigma_{\mathsf{SI}}) = \mathsf{modelOf}(\{\delta_{\mathsf{SI}}\})$*, where* $\delta_{\mathsf{SI}}(\mathcal{G}) = ((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}}) \,;\, \mathsf{RW}_{\mathcal{G}}?)^+$.

(a) Algebraic Laws for Sets of Transactions			
(a.1)	$[\mathcal{T}'] \subseteq \mathsf{Id}$	**(a.2)**	$[\mathcal{T}_1 \cap \mathcal{T}_2] = [\mathcal{T}_1] \mathbin{;} [\mathcal{T}_2]$
(a.3)	$(R_1 \mathbin{;} [\mathcal{T}']) \cap R_2 = (R_1 \cap R_2) \mathbin{;} [\mathcal{T}']$		
(a.4)	$([\mathcal{T}'] \mathbin{;} R_1) \cap R_2 = [\mathcal{T}'] \mathbin{;} (R \cap R_2)$		

(b) Algebraic Laws for (anti)Dependencies	
(b.1)	$\mathsf{WR}(x) \subseteq [\mathsf{Writes}_x] \mathbin{;} \mathsf{WR}(x) \mathbin{;} [\mathsf{Reads}_x]$
(b.2)	$\mathsf{WW}(x) \subseteq [\mathsf{Writes}_x] \mathbin{;} \mathsf{WW}(x) \mathbin{;} [\mathsf{Writes}_x]$
(b.3)	$\mathsf{RW}(x) \subseteq [\mathsf{Reads}_x] \mathbin{;} \mathsf{RW}(x) \mathbin{;} [\mathsf{Writes}_x]$
(b.4)	$\mathsf{WR}(x) \subseteq \mathsf{WR}(x) \setminus \mathsf{Id}$
(b.5)	$\mathsf{WW}(x) \subseteq \mathsf{WW}(x) \setminus \mathsf{Id}$
(b.6)	$\mathsf{RW}(x) \subseteq \mathsf{RW}(x) \setminus \mathsf{Id}$

(c) Universal Algebraic Laws			
(c.1)	$\mathsf{WR}(x) \subseteq \mathsf{VIS}$	**(c.2)**	$\mathsf{WW}(x) \subseteq \mathsf{AR}$
(c.3)	$\mathsf{RW}(x) \subseteq \overline{\mathsf{VIS}^{-1}}$	**(c.4)**	$\mathsf{VIS}^{+} \subseteq \mathsf{VIS}$
(c.5)	$\mathsf{AR}^{+} \subseteq \mathsf{AR}$	**(c.6)**	$\mathsf{VIS} \subseteq \mathsf{AR}$
(c.7)	$[\mathsf{Writes}_x] \mathbin{;} \mathsf{VIS} \mathbin{;} \mathsf{RW}(x) \subseteq \mathsf{AR}$		
(c.8)	$\mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}} \subseteq \overline{\mathsf{VIS}^{-1}}$		
(c.9)	$\overline{\mathsf{VIS}^{-1}} \mathbin{;} \mathsf{VIS} \subseteq \overline{\mathsf{VIS}^{-1}}$		
(c.11)	$(\mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}}) \cap \mathsf{Id} \subseteq \varnothing$		
(c.10)	$(\overline{\mathsf{VIS}^{-1}} \mathbin{;} \mathsf{VIS}) \cap \mathsf{Id} \subseteq \varnothing$		
(c.12)	$\mathsf{AR} \cap \mathsf{Id} \subseteq \varnothing$		

(d) Algebraic Laws induced by the Consistency Guarantee (ρ, π)			
(d.1)	$\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) \subseteq \mathsf{VIS}$	**(d.2)**	$(\pi(\mathsf{VIS}) \mathbin{;} \overline{\mathsf{VIS}^{-1}} \mathbin{;} \rho(\mathsf{VIS})) \setminus \mathsf{Id} \subseteq \mathsf{AR}$
(d.3)	$(\mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \overline{\mathsf{VIS}^{-1}}) \cap \rho(\mathcal{T} \times \mathcal{T})^{-1} \subseteq \overline{\mathsf{VIS}^{-1}}$		
(d.4)	$(\overline{\mathsf{VIS}^{-1}} \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}) \cap \pi(\mathcal{T} \times \mathcal{T})^{-1} \subseteq \overline{\mathsf{VIS}^{-1}}$		

Figure 3 Algebraic laws satisfied by an abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$. Here $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$. The inequalities in part **(d)** are valid under the assumption that $\mathcal{X} \in \mathsf{Executions}(\{\rho, \pi\})$.

Theorem 11(1) has been proved in [23], we proved Theorem 11(3) in [14], and the only if condition of Theorem 11(2) was proved in [6]. However, to the best of our knowledge, the if condition of the Theorem is new in this paper, and leads to a specification of PSI which describes the kind of allowed cycles more precisely than the previous formalisation in [15]. We prove it formally in §5.

4 Algebraic Laws for Weak Consistency

Having two different styles for specifying consistency models gives rise to the following problems:
Weak Correspondence Problem: given a x-specification Σ, determine a non-trivial g-specification Δ which over-approximates Σ, that is such that $\mathsf{modelOf}(\Sigma) \subseteq \mathsf{modelOf}(\Delta)$.
Strong Correspondence Problem: Given a x-specification Σ, determine an equivalent g-specification Δ, that is such that $\mathsf{modelOf}(\Sigma) = \mathsf{modelOf}(\Delta)$.

We first focus on the weak correspondence problem, and we discuss the strong correspondence problem in §5. This problem is not only of theoretical interest. Determining a g-specification Δ that over-approximates a x-specification Σ corresponds to establishing one or more conditions satisfied by all cycles of dependency graphs from the set $\{\mathsf{graph}(\mathcal{X}) \mid \mathcal{X} \in \mathsf{Executions}(\Sigma)\}$. Cycles in a dependency graph that do not respect such a condition are called Σ-*critical*, and graphs that admit a Σ-critical cycle cannot be obtained from abstract executions in $\mathsf{Executions}(\Sigma)$. One can ensure that an application running under the model Σ is *robust*, i.e. it only produces serialisable behaviours, by checking for the absence of non-Σ-critical cycles at static time [6, 16]. Robustness of an application can also be checked at run-time, by incrementally constructing the dependency graph of executions, and detecting the presence of non-Σ-critical cycles [29].

General Methodology. Let Σ be a given x-specification. We tackle the weak correspondence problem in two steps.

First, we identify a set of inequations that hold for all the executions $\mathcal{X}$ satisfying consistency guarantees (ρ, π) in Σ. There are two kinds of such inequations. The first are the inequations in Figure 3, and the second the inequations corresponding to the axioms of the Kleene Algebra $(2^{\mathbb{T} \times \mathbb{T}}, \varnothing, \mathsf{Id}, \cup, ;, \cdot^{*})$ and the boolean algebra $(2^{\mathbb{T} \times \mathbb{T}}, \varnothing, \mathbb{T} \times \mathbb{T}, \cup, \cap, \bar{\cdot})$. The exact meaning of the inequations in Figure 3 is discussed later in this section.

Second, we exploit our inequations to derive inequations of the form $R_{\mathcal{X}} \subseteq \mathsf{AR}_{\mathcal{X}}$ for every

$\mathcal{X} \in \mathsf{Executions}(\Sigma)$. Here $R_{\mathcal{X}}$ is a relation built from dependencies and anti-dependencies in $\mathsf{graph}(\mathcal{X})$, i.e. $R_{\mathcal{X}} \subseteq (\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}} \cup \mathsf{RW}_{\mathcal{X}})^*$. Because $\mathsf{AR}_{\mathcal{X}}$ is acyclic (that is $\mathsf{AR}_{\mathcal{X}}^{+} \cap \mathsf{Id} \subseteq \emptyset$), we may conclude that $R_{\mathcal{X}}$ is acyclic for any $\mathcal{X} \in \mathsf{Executions}(\Sigma)$.

Some of the inequations we develop, namely those in Figure 3(**d**), are parametric in the consistency guarantee (ρ, π). As a consequence, our approach can be specialised to any consistency model that is captured by our framework. To show its applicability, we infer several robustness criteria for the consistency models that we have presented.

Technical Development. Let $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$, and $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$. We now explain the inequations in Figure 3. Among these, the inequations in Figures 3(**a**) and (**b**) should be self-explanatory, as well as many of those in Figure 3(**c**).

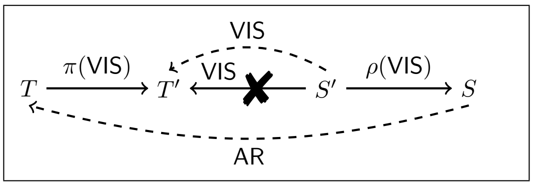
In Figure 3(**c**), inequations **(c.1)**, **(c.2)** and **(c.3)** relate dependencies to visibility and arbitration. The inequation **(c.3)** is non-standard, and it relates anti-dependencies to a novel *anti-visibility* relation $\overline{\mathsf{VIS}^{-1}}$, defined as $T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$ iff $\neg(S \xrightarrow{\mathsf{VIS}} T)$. In words, T is *anti-visible* to S if S does not observe the effects of T. As we will explain later, anti-visibility plays a fundamental role in the development of our algebraic laws.

Proof Sketch of Inequation (c.3). Suppose $T \xrightarrow{\mathsf{RW}(x)} S$ for some object $x \in \mathsf{Obj}$. By definition, there exists a transaction T' such that $T' \xrightarrow{\mathsf{WR}(x)} T$ and $T' \xrightarrow{\mathsf{WW}(x)} S$. In particular, $T' \xrightarrow{\mathsf{AR}} S$ by the inequation **(c.2)**. Now, if it were $S \xrightarrow{\mathsf{VIS}} T$, by the definition of $\mathsf{graph}(\mathcal{X})$ we would have that $S \xrightarrow{\mathsf{WR}(x)} T$, since S would be the AR-supremum of the set of transactions visible to T, and writing to object x. But we already have $T' \xrightarrow{\mathsf{WR}(x)} S$, causing a contradiction. Therefore, $T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$. ◂

Another non-trivial inequation is **(c.7)** in Figure 3(**c**). It says that if a transaction T reads a value for an object x that is later updated by another transaction S ($T \xrightarrow{\mathsf{RW}} S$), then the update of S is more recent (i.e. it follows in arbitration) than all the updates to x seen by T.

The inequations in Figure 3(**d**) are specific to a consistency guarantee (ρ, π), and hold for an execution $\mathcal{X}$ when the execution satisfies (ρ, π). The inequation **(d.1)** is just the definition of consistency guarantee. The next inequation **(d.2)** is where the novel anti-visibility relation, introduced previously, comes into play. While the consistency guarantee (ρ, π) expresses when arbitration induces transactions related by visibility, the inequation **(d.2)** expresses when anti-visibility induces transactions related by arbitration. To emphasise this correspondence, we call the inequation **(d.2)** *co-axiom* induced by (ρ, π). Later in this section, we show how by exploiting the co-axiom induced by several consistency guarantees, we can infer robustness criteria for several consistency models.

Proof of Inequation (d.2). Suppose that $\mathcal{X} \in \mathsf{Executions}(\{(\rho, \pi)\})$. Let $T, T', S', S \in \mathcal{T}$ be such that $T \neq S$, $T \xrightarrow{\rho(\mathsf{VIS})} T' \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S' \xrightarrow{\pi(\mathsf{VIS})} S$. Because AR is total, either $S \xrightarrow{\mathsf{AR}} T$ or $T \xrightarrow{\mathsf{AR}} S$. However, the former case is not possible. If so, we would have $S' \xrightarrow{\rho(\mathsf{VIS})} S \xrightarrow{\mathsf{AR}} T \xrightarrow{\pi(\mathsf{VIS})} T'$. But then because $\mathcal{X} \in \mathsf{Executions}(\{(\rho, \pi)\})$, by the inequation **(d.1)**, it follows that $S' \xrightarrow{\mathsf{VIS}} T'$, contradicting the assumption that $T' \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S'$.



Therefore, it has to be $T \xrightarrow{\mathsf{AR}} S$. The proof of this proposition is depicted in the above figure. Dashed edges represent the consequences of assuming $S \xrightarrow{\mathsf{AR}} T$. The pair $(T', S') \in \overline{\mathsf{VIS}^{-1}}$ is represented by a crossed edge labelled VIS, which connects S' to T', to emphasize the fact that $T' \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S' \iff \neg(S' \xrightarrow{\mathsf{VIS}} T')$. ◂

The last inequations **(d.3)** and **(d.4)** in Figure 3(**d**) show that anti-visibility edges of $\mathcal{X}$ are also induced by the consistency guarantee (ρ, π). We prove them formally in (§C), where we also illustrate some of its applications.

Applications. The algebraic laws of Figure 3 can be used to characterise non-Σ-critical cycles, for any x-specification Σ. Below we give several applications to the consistency models introduced before. Henceforth, we use the notation $R_1 \overset{\textbf{(eq)}}{\subseteq} R_2$ to denote that the inequality $R_1 \subseteq R_2$ follows from the inequation **(eq)**. Due to lack of space, we only give proof sketches of our theorems, and defer complete proofs to Appendix (§C).

▸ **Theorem 12.** *For all* $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SER}})$*, the relation* $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}} \cup \mathsf{RW}_{\mathcal{X}})$ *is acyclic.*

Proof. The co-axiom of the consistency guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{Id}}) \in \Sigma_{\mathsf{SER}}$ is the inequation $\overline{\mathsf{VIS}_{\mathcal{X}}^{-1}}\backslash\mathsf{Id} \subseteq \mathsf{AR}_{\mathcal{X}}$. Together with the inequation **(c.3)**, this leads to $\mathsf{RW}_{\mathcal{X}} \overset{\textbf{(b.6)}}{\subseteq} \mathsf{RW}_{\mathcal{X}}\backslash\mathsf{Id} \overset{\textbf{(c.3)}}{\subseteq} \overline{\mathsf{VIS}_{\mathcal{X}}^{-1}}\backslash\mathsf{Id} \overset{\textbf{(d.2)}}{\subseteq} \mathsf{AR}$. From here it is easy to infer that $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}} \cup \mathsf{RW}_{\mathcal{X}}) \subseteq \mathsf{AR}_{\mathcal{X}}$, hence the former is acyclic. ◂

▸ **Theorem 13.** *For all* $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SI}})$*, every cycle in* $\mathsf{graph}(\mathcal{X})$ *always includes two consecutive* $\mathsf{RW}_{\mathcal{X}}$ *edges. That is,* $((\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}}) \,;\, \mathsf{RW}_{\mathcal{X}}?)$ *is acyclic.*

Proof Sketch. Let $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SI}})$. We argue that, for any execution $\mathcal{X} \in \mathsf{SI}$, the following inequations can be derived in an algebraic manner: **(i)** $\mathsf{WW}_{\mathcal{X}} \subseteq \mathsf{VIS}_{\mathcal{X}}$, and **(ii)** $\mathsf{VIS}_{\mathcal{X}} \,;\, \mathsf{RW}_{\mathcal{X}}? \subseteq \mathsf{AR}_{\mathcal{X}}$. Inequation **(i)** is a consequence of SI enforcing the write-conflict detection property, i.e. $(\rho_x, \rho_x) \in \Sigma_{\mathsf{SI}}$ for any object $x \in \mathsf{Obj}$:

$$\mathsf{WW}(x) \overset{\textbf{(b.2)}}{\subseteq} [\mathsf{Writes}_x] \,;\, \mathsf{WW}(x) \,;\, [\mathsf{Writes}_x] \overset{\textbf{(c.2)}}{\subseteq} [\mathsf{Writes}_x] \,;\, \mathsf{AR}_{\mathcal{X}} \,;\, [\mathsf{Writes}_x] \overset{\textbf{(d.1)}}{\subseteq} \mathsf{VIS}_{\mathcal{X}}.$$

Inequation **(ii)** is a consequence of the co-axiom $(\mathsf{VIS}_{\mathcal{X}} \,;\, \mathsf{RW}_{\mathcal{X}})\backslash\mathsf{Id} \subseteq \mathsf{AR}$ induced by the consistency guarantee $(\rho_{\mathsf{Id}}, \rho_S)$ (as we show in (§D), the latter can be strengthened to $(\mathsf{VIS}_{\mathcal{X}} \,;\, \mathsf{RW}_{\mathcal{X}}) \subseteq \mathsf{AR}_{\mathcal{X}}$), and of inequation **(c.3)**. As a consequence of **(i)**, and Equation **(c.1)**, it follows that $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}}) \subseteq \mathsf{VIS}_{\mathcal{X}}$. By substituting $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})$ for $\mathsf{VIS}_{\mathcal{X}}$ in **(ii)**, we obtain that $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}}) \,;\, \mathsf{RW}_{\mathcal{X}}?$ is included in AR, hence it is acyclic. ◂

▸ **Theorem 14.** *For all* $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$*, it is not possible that all anti-dependencies in a cycle of* $\mathsf{graph}(\mathcal{X})$ *are over the same object:* $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})^* \,;\, \mathsf{RW}(x)$ *is acyclic for all* $x \in \mathsf{Obj}$.

Proof Sketch. Let $\mathcal{X} \in \Sigma_{\mathsf{PSI}}$. Here we prove a slightly weaker result, and defer the full proof of Theorem 14 to (§C). We show that, for any $x \in \mathsf{Obj}$, the relation $([\mathsf{Writes}_x] \,;\, (\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})^* \,;\, \mathsf{RW}_{\mathcal{X}}(x))$ is acyclic. The proof relies on the following three properties of $\mathcal{X}$, each of which can be proved using the laws of Figure 3: **(i)** $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})^+ \subseteq \mathsf{VIS}_{\mathcal{X}}$, **(ii)** $([\mathsf{Writes}_x] \,;\, \mathsf{RW}_{\mathcal{X}}) \subseteq \mathsf{AR}_{\mathcal{X}}$, and **(iii)** $[\mathsf{Writes}_x] \,;\, \mathsf{VIS} \,;\, \mathsf{RW}_{\mathcal{X}}(x) \subseteq \mathsf{AR}_{\mathcal{X}}$. The latter is an immediate consequence of Inequation **(c.7)**. Inequation **(i)** can be proved as in Theorem 13, since $(\rho_x, \rho_x) \in \mathsf{PSI}$ for any object $x \in \mathsf{Obj}$. Inequation **(ii)** is a consequence of the co-axiom induced by (ρ_x, ρ_x):

$$\begin{aligned}[\mathsf{Writes}_x] \,;\, \mathsf{RW} &\overset{\textbf{(b.6)}}{\subseteq} [\mathsf{Writes}_x] \,;\, (\mathsf{RW}_{\mathcal{X}}(x)\backslash\mathsf{Id}) \overset{\textbf{(a.4)}}{=} ([\mathsf{Writes}_x] \,;\, \mathsf{RW}_{\mathcal{X}}(x))\backslash\mathsf{Id} \overset{\textbf{(b.3)}}{\subseteq} \\ &([\mathsf{Writes}_x] \,;\, \mathsf{RW}_{\mathcal{X}}(x) \,;\, [\mathsf{Writes}_x])\backslash\mathsf{Id} \overset{\textbf{(c.3)}}{\subseteq} ([\mathsf{Writes}_x] \,;\, \overline{\mathsf{VIS}_{\mathcal{X}}^{-1}} \,;\, [\mathsf{Writes}_x])\backslash\mathsf{Id} \overset{\textbf{(d.2)}}{\subseteq} \mathsf{AR}_{\mathcal{X}}.\end{aligned}$$

By combining **(i)**, **(ii)** and **(iii)**, we obtain that $[\mathsf{Writes}_x] \,;\, (\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})^* \,;\, \mathsf{RW}_{\mathcal{X}}(x) \subseteq \mathsf{AR}_{\mathcal{X}}$, and therefore $[\mathsf{Writes}_x] \,;\, (\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}})^* \,;\, \mathsf{RW}_{\mathcal{X}}(x)$ is acyclic. ◂

▸ **Theorem 15.** *Let* $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{RB}})$*. Say that a* $\mathsf{RW}_{\mathcal{X}}$ *edge in a cycle of* $\mathsf{graph}(\mathcal{X})$ *is* serialised *if its endpoints are connected to serialisable transactions via a sequence of* $\mathsf{WR}_{\mathcal{X}}$ *edges. Then all cycles in* $\mathsf{graph}(\mathcal{X})$ *have at least one non-serialised* $\mathsf{RW}_{\mathcal{X}}$ *edge. Formally, let* $\Vdash\mathsf{RW}_{\mathcal{X}}\dashv\Vert$ *be* $([\mathtt{SerTx}] \,;\, (\mathsf{WR}_{\mathcal{X}})^* \,;\, \mathsf{RW}_{\mathcal{X}} \,;\, (\mathsf{WR}_{\mathcal{X}})^* \,;\, [\mathtt{SerTx}])$*. Then* $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}} \cup \Vdash\mathsf{RW}_{\mathcal{X}}\dashv\Vert)$ *is acyclic.*

Proof Sketch. Let $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{RB}})$. Note that by equations **(c.1)**, **(c.3)**, **(c.8)** and **(c.9)**, we can easily deduce that $\mathsf{WR}_{\mathcal{X}}^* \,;\, \mathsf{RW}_{\mathcal{X}} \,;\, \mathsf{WR}_{\mathcal{X}}^* \subseteq \overline{\mathsf{VIS}_{\mathcal{X}}^{-1}}$. The co-axiom of the consistency guarantee (ρ_S, ρ_S), gives $(\Vdash\mathsf{RW}_{\mathcal{X}}\dashv\Vert)\backslash\mathsf{Id} \subseteq ([\mathtt{SerTx}] \,;\, \overline{\mathsf{VIS}_{\mathcal{X}}^{-1}} \,;\, [\mathsf{Writes}_x])\backslash\mathsf{Id} \subseteq \mathsf{AR}_{\mathcal{X}}$. In practice, we can prove a stronger inequation, namely $\Vdash\mathsf{RW}_{\mathcal{X}}\dashv\Vert \subseteq \mathsf{AR}_{\mathcal{X}}$. From here it is immediate to deduce that $(\mathsf{WR}_{\mathcal{X}} \cup \mathsf{WW}_{\mathcal{X}} \cup \Vdash\mathsf{RW}_{\mathcal{X}}\dashv\Vert) \subseteq \mathsf{AR}$, from which the result follows.

$$\left\{\begin{array}{llllll}
\mathsf{WR} \subseteq X_V & (\text{V1}) & X_V \,;\, X_V \subseteq X_V & (\text{V2}) & \displaystyle\bigcup_{\{x \mid (\rho_x,\rho_x)\in\Sigma\}} \mathsf{WW}(x) \subseteq X_V & (\text{V3}) \\
 & & & & \rho(X_V)\,;\,X_A\,;\,\pi(X_V) \subseteq X_V & (\text{V4}) \\
\mathsf{WW} \subseteq X_A & (\text{A1}) & X_V \subseteq X_A & (\text{A2}) & \displaystyle\bigcup_{x\in\mathsf{Obj}} ([\mathsf{Writes}_x]\,;\,X_V\,;\,\mathsf{RW}(x)) \subseteq X_A & (\text{A3}) \\
 & & X_A \,;\, X_A \subseteq X_A & (\text{A4}) & (\pi(X_V)\,;\,X_N\,;\,\rho(X_V)) \setminus \mathsf{Id} \subseteq X_A & (\text{A5}) \\
\mathsf{RW} \subseteq X_N & (\text{N1}) & X_V \,;\, X_N \subseteq X_N & (\text{N2}) & X_N \,;\, X_V \subseteq X_N & (\text{N3})
\end{array}\right.$$

Figure 4 The system of inequations $\mathsf{System}_\Sigma(\mathcal{G})$ for the simple consistency model Σ and the dependency graph $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$.

5 Characterisation of Simple Consistency Models

We now turn our attention to the *Strong Correspondence Problem* presented in §4. Given a x-specification $\Sigma = \{(\rho_1, \pi_1), \cdots, (\rho_n, \pi_n)\}$ and a dependency graph $\mathcal{G}$, we want to find out a sufficient and necessary condition for determining whether $\mathcal{G} = \mathsf{graph}(\mathcal{X})$ for some $\mathcal{X} \in \mathsf{Executions}(\Sigma)$.

In this section we propose a proof technique for solving the strong correspondence problem. This technique applies to a particular class of x-specifications, which we call *simple* x-specifications. This class includes several of the consistency models we have presented. We conclude the section by showing why our proof technique is not sound in the case of non-simple x-specifications.

Characterisation of Simple x-specifications. Recall that for each object $x \in \mathsf{Obj}$, the function ρ_x of an abstract execution $\mathcal{X}$ maps a relation R on transactions in $\mathcal{X}$ to the relation $[\mathsf{Writes}_x]$.

▸ **Definition 16.** A x-specification Σ is *simple* if there exists a consistency guarantee (ρ, π) such that $\Sigma \subseteq \{(\rho, \pi)\} \cup \{(\rho_x, \rho_x)\}_{x\in\mathsf{Obj}}$.

That is, a simple x-specification Σ contains at most one consistency guarantee, beside those of the form (ρ_x, ρ_x) which express the write-conflict detection for some object $x \in \mathsf{Obj}$. Recall that the axiom induced by (ρ_x, ρ_x) for an abstract execution $\mathcal{X}$ is $([\mathsf{Writes}_x] \,;\, \mathsf{AR}_\mathcal{X} \,;\, [\mathsf{Writes}_x]) \subseteq \mathsf{VIS}_\mathcal{X}$. Among the x-specifications that we have presented in this paper, the only non-simple one is $\Sigma_{\mathsf{SI+SER}}$.

For simple x-specifications, it is possible to solve the strong correspondence problem, as illustrated by the main result of this section:

▸ **Theorem 17.** *Let $\Sigma \subseteq \{(\rho, \pi)\} \cup \{(\rho_x, \rho_x) \mid x \in \mathsf{Obj}\}$ be a simple x-specification, and let $\mathcal{G} \in \mathsf{Graphs}$. Define $\mathsf{System}_\Sigma(\mathcal{G})$ to be the collection of inequations about X_V, X_A and X_N depicted in Figure 4; the inequations involving (ρ, π) are included in $\mathsf{System}_\Sigma(\mathcal{G})$ only if (ρ, π) is in Σ. Let $(X_V = \mathsf{VIS}_0, X_A = \mathsf{AR}_0, X_N = \mathsf{AntiVIS}_0)$ be the smallest solution of $\mathsf{System}_\Sigma(\mathcal{G})$.*[4] *Note that AR_0 need not to be total here. Then, $\mathcal{G} = \mathsf{graph}(\mathcal{X})$ for some $\mathcal{X} \in \mathsf{Executions}(\Sigma)$ iff $\mathsf{AR}_0 \cap \mathsf{Id} = \varnothing$.*

As an immediate consequence of Theorem 17 we obtain the following result:

▸ **Corollary 18.** *Let $\Sigma \subseteq \{(\rho, \pi)\} \cup \{(\rho_x, \rho_x) \mid x \in \mathsf{Obj}\}$ be a simple x-specification. For each dependency graph $\mathcal{G}$, let $(X_V = \_, X_A = \mathsf{AR}_\mathcal{G}, X_N = \_)$ be the smallest solution of $\mathsf{System}_\Sigma(\mathcal{G})$, and define $\delta(\mathcal{G}) = \mathsf{AR}_\mathcal{G}$. Then $\mathsf{modelOf}(\Sigma) = \mathsf{modelOf}(\{\delta\})$.*

In the rest of the section, we will discuss the if direction of Theorem 17. Assume a simple x-specification $\Sigma \subseteq \{(\rho, \pi)\} \cup \{(\rho_x, \rho_x)\}_{x\in\mathsf{Obj}}$, and a dependency graph $\mathcal{G}$ with the property that the smallest solution $(X_V = \mathsf{VIS}_0, X_A = \mathsf{AR}_0, X_N = \mathsf{AntiVis}_0)$ of $\mathsf{System}_\Sigma(\mathcal{G})$ is such that AR_0 is irreflexive. We will construct construct an abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR}) \in \mathsf{Executions}(\Sigma)$

[4] That is, for any other solution $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ we have $\mathsf{VIS}_0 \subseteq \mathsf{VIS}, \mathsf{AR}_0 \subseteq \mathsf{AR}$ and $\mathsf{AntiVIS}_0 \subseteq \mathsf{AntiVIS}$.

with $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$ in two steps.

First, we note that for any solution $(X_V = \mathsf{VIS}', X_A = \mathsf{AR}', X_N = \mathsf{AntiVIS}')$ of $\mathsf{System}_{\Sigma}(\mathcal{G})$ with irreflexive AR', we may take its parts and construct the triple $(\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}', \mathsf{AR}')$. This tuple satisfies almost all the properties required for being an abstract execution in $\mathsf{Executions}(\Sigma)$ except that AR' does not necessarily relate all the transactions in $\mathcal{T}_{\mathcal{G}}$. It is an example of *pre-execution*:

▸ **Definition 19.** A *pre-execution* $\mathcal{P} = (\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}, \mathsf{AR})$ is a tuple that satisfies all the constraints of abstract executions, except that AR is not necessarily total, although AR is still required to be total over the set Writes_x for every object x.

The notation adopted for abstract executions naturally extends to pre-executions; also, for any valid pre-execution $\mathcal{P}$, $\mathsf{graph}(\mathcal{P})$ is well defined. Given a x-specification Σ, we let $\mathsf{PreExecutions}(\Sigma)$ be the set of all valid pre-executions that satisfy the consistency guarantees $(\rho, \pi) \in \Sigma$.

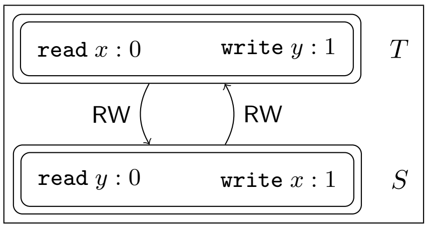
▸ Proposition 20. Let $(X_V = \mathsf{VIS}', X_A = \mathsf{AR}', X_N = \mathsf{AntiVIS}')$ be a solution to $\mathsf{System}_{\Sigma}(\mathcal{G})$. If $\mathsf{AR} \cap \mathsf{Id} = \varnothing$, then $\mathcal{P} = (\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}', \mathsf{AR}') \in \mathsf{PreExecutions}(\Sigma)$; moreover, $\mathsf{graph}(\mathcal{P}) = \mathcal{G}$.

Proof Sketch. Each of the inequations (V1)-(V4) and (A1)-(A4) enforces one particular property required by elements of $\mathsf{PreExecutions}(\Sigma)$:

- The inequations (A1),(A2) and (A4) ensure that $\mathcal{P}$ is a pre-execution; in particular the first of these inequations ensures that AR' is a total relation over any of the sets Writes_x, $x \in \mathsf{Obj}$; because of the inequation (V2), $\mathcal{P}$ respects causality;
- The inequations (V1), (A1) and (A3) enforce the Last Write Wins policy, a statement which we prove formally in (§D); in particular, the inequation (A3) prevents a transaction T reading a value for object x, from seeing any transaction writing a newer value for such an object: if $S \xrightarrow{\mathsf{WR}(x)} V \xrightarrow{\mathsf{RW}(x)} T$, then $S \xrightarrow{\mathsf{AR}'} T$ by inequations (V1) and (A3), which leads to $\neg(T \xrightarrow{\mathsf{VIS}'} S)$ because the inequation (A2) implies $\mathsf{VIS}' \subseteq \mathsf{AR}'$, and $\mathsf{AR}' \cap \mathsf{Id} = \varnothing$ by hypothesis;
- The inequation (V3) implies that the pre-execution $\mathcal{P}$ satisfies each of the consistency guarantees $(\rho_x, \rho_x) \in \Sigma$; finally, the inequation (V4) ensures that the additional consistency guarantee $(\rho, \pi) \in \Sigma$ is also satisfied by $\mathcal{P}$. ◂

Second, we show that some solution $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ of $\mathsf{System}_{\Sigma}(\mathcal{G})$ has a total AR. This implies the if direction of Theorem 17. Recall our assumption that the smallest solution $(X_V = \mathsf{VIS}_0, X_A = \mathsf{AR}_0, X_N = \mathsf{AntiVIS}_0)$ of $\mathsf{System}_{\mathcal{G}}(\Sigma)$ is such that $\mathsf{AR}_0 \cap \mathsf{Id} = \varnothing$. This requirement is necessary for the solution $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ above to exist. However, it is not the only necessary condition. The following example shows that the smallest solution of $\mathsf{System}_{\Sigma}(\mathcal{G})$ also has to satisfy inequations (A5) and (N1)-(N3).

Example. Consider the simple x-specification Σ_{RB}. We define the system $\mathsf{System}^{\mathsf{bad}}_{\Sigma_{\mathsf{RB}}}(\mathcal{G})$ by removing inequations (A5), (N1), (N2), and (N3) from $\mathsf{System}_{\Sigma_{\mathsf{RB}}}(\mathcal{G})$. This system has two unknowns X_V, X_A. Consider the dependency graph depicted to the right. Here the transactions T, S have a double border to denote that they have been marked as serialisable: $T, S \ni \mathtt{SerTx}$. We also omitted a transaction T_0 that is visible from T, S and writes the initial value 0 for objects x and y. The smallest solution to $\mathsf{System}^{\mathsf{bad}}_{\Sigma_{\mathsf{RB}}}(\mathcal{G})$ is given by $(X_V = \mathsf{VIS}_0', X_A = \mathsf{AR}_0')$, where $\mathsf{VIS}_0' = \mathsf{AR}_0' = \{(T_0, T), (T_0, S)\}$. AR_0' is irreflexive, but it is immediate to observe that $\mathcal{T}_{\mathcal{G}} \notin \mathsf{modelOf}(\Sigma_{\mathsf{RB}})$.



On the other hand, let $(X_V = \mathsf{VIS}_0, X_A = \mathsf{AR}_0, X_N = \mathsf{AntiVIS}_0)$ be the smallest solution of the original System $\mathsf{System}_{\Sigma_{\mathsf{RB}}}(\mathcal{G})$. In this case we have that $\mathsf{AntiVIS}_0 = \{(T, S), (S, T)\}$ because of the inequation (N1), and therefore $(T, T) \in \mathsf{AR}_0$ because of the inequations (A5) and (A4). It follows that AR_0 is not irreflexive. ◂

It turns out that the conditions we have placed on $\mathsf{System}_{\Sigma}(\mathcal{G})$, as well as our initial assumption that its smallest solution $(X_V = \_, X_A = \mathsf{AR}_0, X_N = \_)$ is such that AR_0 is irreflexive, are also sufficient to establish the existence of another solution $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ such that AR is a total relation.

To prove this claim, we take an incremental approach. Initially, let $n = 0$, and consider the solution $(X_V = \mathsf{VIS}_n, X_A = \mathsf{AR}_n, X_N = \mathsf{AntiVIS}_n)$ of $\mathsf{System}_{\mathcal{G}}$. If all different T, S are related by AR_n, then AR_n is a strict total order, and we can take $\mathsf{VIS} := \mathsf{VIS}_n, \mathsf{AR} := \mathsf{AR}_n, \mathsf{AntiVIS} := \mathsf{AntiVIS}_n$. Otherwise, we choose two transactions T_n, S_n that are not related by AR_n. We compute the smallest solution $(X_V = \mathsf{VIS}_{n+1}, X_A = \mathsf{AR}_{n+1}, X_N = \mathsf{AntiVIS}_{n+1})$ to $\mathsf{System}_{\Sigma}(\mathcal{G})$ with the additional property that $\mathsf{AR}_{n+1} \supseteq \mathsf{AR}_n \cup \{(T_n, S_n)\}$, and we repeat the procedure for $n := n + 1$.

Interestingly enough, the solution $(X_V = \mathsf{VIS}_{n+1}, X_A = \mathsf{AR}_{n+1}, X_N = \mathsf{AntiVIS}_{n+1})$ in the procedure above can be expressed in function of $\mathsf{AR}_n, \mathsf{VIS}_n, \mathsf{AntiVIS}_n$. For example, AR_{n+1} can be expressed as follows:

▸ Proposition 21. Let $\Delta\mathsf{AR}_n := \mathsf{AR}_n? \; ; \{(T_n, S_n)\} \; ; \mathsf{AR}_n?$. Then $\mathsf{AR}_{n+1} = \mathsf{AR}_n \cup \Delta\mathsf{AR}_n$.

An important consequence of Proposition 21 is that the acyclicity of the $\{\mathsf{AR}_i\}_{i=0}^{n}$ relations is preserved by the procedure described above.

▸ **Corollary 22.** *If* $\mathsf{AR}_n \cap \mathsf{Id} = \varnothing$*, then* $\mathsf{AR}_{n+1} \cap \mathsf{Id} = \varnothing$.

Proof. Because $\mathsf{AR}_n \cap \mathsf{Id} = \varnothing$ by hypothesis, by Proposition 21 it suffices to show that $\Delta\mathsf{AR}_n$ is irreflexive: if $(T, T) \in \Delta\mathsf{AR}_n$ for some $T \in \mathcal{T}_{\mathcal{G}}$, then it must be $T \xrightarrow{\mathsf{AR}_n?} T_n$ and $S_n \xrightarrow{\mathsf{AR}_n?} T$, from which it follows that $S_n \xrightarrow{\mathsf{AR}_n?} T_n$. But this contradicts the hypothesis that AR_n does not relate transactions T_n and S_n. Therefore, $(T, T) \notin \Delta\mathsf{AR}_n$ for any $T \in \mathcal{T}_{\mathcal{G}}$, i.e. $\Delta\mathsf{AR}_n$ is irreflexive. ◂

We have now everything in place to prove Theorem 17.

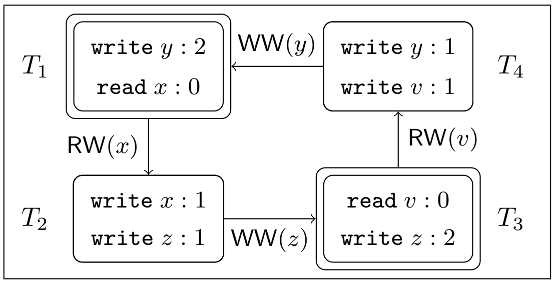
Proof of Theorem 17. For the only if direction, let $\mathcal{X} = (\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}, \mathsf{AR}) \in \mathsf{Executions}(\Sigma)$, and suppose that $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$. It follows from the algebraic laws developed in Section 4 that $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \overline{\mathsf{VIS}^{-1}})$ is a solution to $\mathsf{System}_{\mathcal{G}}(\Sigma)$. Let $(X_V = \mathsf{VIS}_0, X_A = \mathsf{AR}_0, X_N = \mathsf{AntiVIS}_0)$ be the smallest solution to $\mathsf{System}_{\Sigma}(\mathcal{G})$. We have that $\mathsf{AR}_0 \subseteq \mathsf{AR}$, and because AR is irreflexive, so is AR_0.

Let now $\mathcal{G}$ be a dependency graph, and let $(X_V = \_, X_A = \mathsf{AR}_0, X_N = \_)$ be the smallest solution to $\mathsf{System}_{\Sigma}(\mathcal{G})$. Assume that $\mathsf{AR}_0 \cap \mathsf{Id} = \varnothing$. If AR_0 is a strict total order, then we have that $\mathcal{X} = (\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}_0, \mathsf{AR}_0) \in \mathsf{Executions}(\Sigma)$, and $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$, by Proposition 20 and we are done. Otherwise, we choose (T_0, S_0) such that neither $T_0 = S_0$, $T_0 \xrightarrow{\mathsf{AR}_0} S_0$, nor $S_0 \xrightarrow{\mathsf{AR}_0} T_0$. We compute the smallest solution $(X_V = \_, X_A = \mathsf{AR}_1, X_N = \_)$ of $\mathsf{System}_{\Sigma}(\mathcal{G})$ with the property that $\mathsf{AR}_1 \supseteq \mathsf{AR}_0 \cup \{(T_0, S_0)\}$: by Corollary 22 it follows that AR_1 is irreflexive. We iterate this procedure until we do not reach a solution $(X_V = \_, X_A = \mathsf{AR}_n, X_N = \_)$ such that AR_n is total; such a solution exists because at each step of the iteration we decrease the number of transactions that are not related by AR_i, $i = 1, \cdots, n$. By Corollary 22 we know that $\mathsf{AR}_n \cap \mathsf{Id} = \varnothing$, hence by Proposition 20 we obtain that $(\mathcal{T}_{\mathcal{G}}, \mathsf{VIS}_n, \mathsf{AR}_n) \in \mathsf{PreExecutions}(\Sigma)$. Since AR_n is a total relation, it follows that $\mathcal{P} \in \mathsf{Executions}(\Sigma)$, as we wanted to prove. ◂

Applications. Corollary 18 is a powerful proof technique for solving the strong correspondence problem. Given a simple x-specification Σ, we can recover an equivalent g-specification Δ by simply solving the collection of systems $\mathsf{System}_{\Sigma}(\cdot)$. Below we give a proof of Theorem 11.

Proof Sketch of Theorem 11. For each of the x-specifications Σ considered in Theorem 11, and for any dependency graph $\mathcal{G}$, we compute the smallest solution $(X_V = \mathsf{VIS}_{\mathcal{G}}^{\Sigma}, X_A = \mathsf{AR}_{\mathcal{G}}^{\Sigma}, X_N = \mathsf{AntiVIS}_{\mathcal{G}}^{\Sigma})$ of $\mathsf{System}_{\Sigma}(\mathcal{G})$, and note that $\mathsf{AR}_{\mathcal{G}}^{\Sigma}$ coincides with the union of the desired relations $\delta(\mathcal{G})$. For example, for Σ_{PSI}, we obtain that $\mathsf{VIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}} = (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{+}$, $\mathsf{AntiVIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}} = \mathsf{VIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}}? \; ; \mathsf{RW}_{\mathcal{G}} \; ; \mathsf{VIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}}?$, while $\mathsf{AR}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}} = \mathsf{VIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}} \cup \left(\bigcup_{x \in \mathsf{Obj}} (\mathsf{VIS}_{\mathcal{G}}^{\Sigma_{\mathsf{PSI}}})? \; ; \mathsf{RW}(x)\right)^{+} = \delta_{\mathsf{PSI}_0}(\mathcal{G}) \cup \left(\bigcup_{x \in \mathsf{Obj}} \delta_{\mathsf{PSI}(x)}(\mathcal{G})\right)$. We show how to solve the system of inequations in (§D). ◂

Incompleteness for Arbitrary x-specifications of Consistency Models. One could ask whether Theorem 17 holds for non-simple x-specifications Σ, where $\mathsf{System}_\Sigma(\mathcal{G})$ is defined by including inequations of the form (V4), (A5), for each consistency guarantee $(\rho, \pi) \in \Sigma$. Unfortunately, this is not the case. Consider the x-specification $\Sigma = \{(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}}), (\rho_S, \rho_S)\}$, and let $\mathcal{G}$ be the dependency graph depicted to the right. Recall that transactions with a double border are marked as serialisable. We omitted from $\mathcal{G}$ a transaction T_0 which writes the value 0 for objects x, v, and which is seen by T_1, T_3. For the dependency graph $\mathcal{G}$, the least solution of $\mathsf{System}_\Sigma(\mathcal{G})$ is $(X_V = \_, X_A = \mathsf{AR}_0, X_N = \_)$, where $\mathsf{AR}_0 = \{(T_2, T_3), (T_4, T_1)\} \cup \{(T_0, T_i)\}_{i=1}^{4}$. That is, AR_0 is acyclic. However, there exists no abstract execution $\mathcal{X} \in \mathsf{Executions}(\Sigma)$ such that $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$. In fact, if such $\mathcal{X}$ existed, then T_1 and T_3 should be related by $\mathsf{AR}_\mathcal{X}$. However, it cannot be $T_1 \xrightarrow{\mathsf{AR}_\mathcal{X}} T_3$: the axiom of the consistency guarantee (ρ_S, ρ_S), $[\mathtt{SerTx}] ; \mathsf{AR}_\mathcal{X} ; [\mathtt{SerTx}] \subseteq \mathsf{VIS}_\mathcal{X}$, would imply $T_1 \xrightarrow{\mathsf{VIS}_\mathcal{X}} T_3$; together with $T_3 \xrightarrow{\mathsf{RW}_\mathcal{X}} T_4$ and the co-axiom induced by $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$, $(\mathsf{VIS}_\mathcal{X} ; \overline{\mathsf{VIS}_\mathcal{X}^{-1}})\setminus\mathsf{Id} \subseteq \mathsf{AR}_\mathcal{X}$, this would mean that $T_1 \xrightarrow{\mathsf{AR}_\mathcal{X}} T_4$. But we also have $T_4 \xrightarrow{\mathsf{AR}_\mathcal{X}} T_1$, hence a contradiction. Similarly, we can prove $\neg(T_3 \xrightarrow{\mathsf{AR}_\mathcal{X}} T_1)$.



6 Conclusion

We have explored the connection between two different styles of specifications for weak consistency models at an algebraic level.We have proposed several laws which we applied to devise several robustness criteria for consistency models. To the best of our knowledge, this is the first generic proof technique for proving robustness criteria of weak consistency models. We have shown that, for a particular class of consistency models, our algebraic approach leads to a precise characterisation of consistency models in terms of dependency graphs. In the future, we intend to mechanise our proof technique for inferring robustness criteria. We also plan to continue exploring the space of algebraic laws that connect dependency graphs to abstract execution: our goal, in this respect, is that of giving a precise characterisation for all consistency models, in terms of dependency graphs.

Related Work. Abstract executions have been introduced by Burckhardt in [10] to model the behaviour of eventually consistent data-stores; They have been used to capture the behaviour of replicated data types [Gotsman et al., 11], geo-replicated databases [Cerone et al., 13] and non-transactional distributed storage systems [Viotti et al., 28].

Dependency graphs have been introduced by Adya [2]; they have been used since to reason about programs running under weak consistency models. Bernardi et al., used dependency graphs to derive robustness criteria of several consistency models [6], including PSI and red-blue; in contrast with our work, the proofs there contained do not rely on a general technique. Also, the proposed criterion for red-blue is less precise than ours. Brutschy et al. generalised the notion of dependency graphs to replicated data types, and proposed a robustness criterion for eventual consistency [8].

Weak consistency also arises in the context of shared memory systems [3]. Alglave et al., proposed the CAT language for specifying weak memory models in [3], which also specifies weak memory models as a set of irreflexive relations over data-dependencies of executions. Castellan [12], and Jeffrey et al. [18], proposed different formalisations of weak memory models via event structures.

The strong correspondence problem (§5) is also highlighted by Bouajjani et al. in [7]: there the authors emphasize the need for general techniques to identify all the *bad patterns* that can arise in dependency-graphs like structures. We solved the strong correspondence problem for SI in [14].

References

1 Microsoft sql server documentation, set transaction isolation level. https://docs.microsoft.com/en-us/sql/t-sql/statements/set-transaction-isolation-level-transact-sql.

2 A. Adya. Weak consistency: A generalized theory and optimistic implementations for distributed transactions. PhD thesis, MIT, 1999.

3 J. Alglave, L. Maranget, and M. Tautschnig. Herding cats: Modelling, simulation, testing, and data mining for weak memory. *ACM Trans. Program. Lang. Syst.*, 36(2):7:1–7:74, 2014.

4 P. Bailis, A. Fekete, A. Ghodsi, J. M. Hellerstein, and I. Stoica. Scalable atomic visibility with RAMP transactions. In *SIGMOD*, 2014.

5 H. Berenson, P. Bernstein, J. Gray, J. Melton, E. O'Neil, and P. O'Neil. A critique of ANSI SQL isolation levels. In *SIGMOD*, 1995.

6 G. Bernardi and A. Gotsman. Robustness against consistency models with atomic visibility. In *27th International Conference on Concurrency Theory, CONCUR 2016, August 23-26, 2016, Québec City, Canada*, pages 7:1–7:15, 2016.

7 A. Bouajjani, C. Enea, R. Guerraoui, and J. Hamza. On verifying causal consistency. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages, POPL 2017, Paris, France, January 18-20, 2017*, pages 626–638, 2017.

8 L. Brutschy, D. Dimitrov, P. Müller, and M. Vechev. Serializability for eventual consistency: Criterion, analysis and applications. In *Proceedings of the 44th ACM SIGPLAN Symposium on Principles of Programming Languages (POPL)*. ACM, January 2017.

9 S. Burckhardt. Principles of eventual consistency. *Found. Trends Program. Lang.*, 1(1-2):1–150, Oct. 2014.

10 S. Burckhardt, M. Fahndrich, D. Leijen, and M. Sagiv. Eventually consistent transactions. In *Proceedings of the 22n European Symposium on Programming (ESOP)*. Springer, March 2012.

11 S. Burckhardt, A. Gotsman, H. Yang, and M. Zawirski. Replicated data types: specification, verification, optimality. In *POPL*, 2014.

12 S. Castellan. Weak memory models using event structures. In *Vingt-septièmes Journées Francophones des Langages Applicatifs (JFLA 2016)*, 2016.

13 A. Cerone, G. Bernardi, and A. Gotsman. A framework for transactional consistency models with atomic visibility. In *CONCUR*. Dagstuhl, 2015.

14 A. Cerone and A. Gotsman. Analysing snapshot isolation. In *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*, PODC '16, pages 55–64, New York, NY, USA, 2016. ACM.

15 A. Cerone, A. Gotsman, and H. Yang. Transaction chopping for parallel snapshot isolation. In *DISC*, 2015.

16 A. Fekete, D. Liarokapis, E. O'Neil, P. O'Neil, and D. Shasha. Making snapshot isolation serializable. *ACM Trans. Database Syst.*, 30(2), 2005.

17 A. Gotsman and H. Yang. Composite replicated data types. In J. Vitek, editor, *Programming Languages and Systems: 24th European Symposium on Programming, ESOP 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015, Proceedings*, pages 585–609, Berlin, Heidelberg, 2015. Springer Berlin Heidelberg.

18 A. Jeffrey and J. Riely. On thin air reads towards an event structures model of relaxed memory. In *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 759–767, 2016.

19 S. Jorwekar, A. Fekete, K. Ramamritham, and S. Sudarshan. Automating the detection of snapshot isolation anomalies. In *VLDB*, 2007.

20 D. Kozen and F. Smith. Kleene algebra with tests: Completeness and decidability. In *Proc. 10th Int. Workshop Computer Science Logic (CSL'96), volume 1258 of Lecture Notes in Computer Science*, pages 244–259. Springer-Verlag, 1996.

21 C. Li, D. Porto, A. Clement, J. Gehrke, N. Preguiça, and R. Rodrigues. Making geo-replicated systems fast as possible, consistent when necessary. In *Presented as part of the 10th USENIX Symposium on Operating Systems Design and Implementation (OSDI 12)*, pages 265–278, Hollywood, CA, 2012. USENIX.

22 W. Lloyd, M. J. Freedman, M. Kaminsky, and D. G. Andersen. Don't settle for eventual: scalable causal consistency for wide-area storage with COPS. In *SOSP*, 2011.

23 C. Papadimitriou. The serializability of concurrent database updates. *Journal of the ACM (JACM)*, 26(4):631–653, 1979.

24 M. Saeida Ardekani, P. Sutra, and M. Shapiro. Non-monotonic snapshot isolation: Scalable and strong consistency for geo-replicated transactional systems. In *SRDS*, 2013.

25 D. Shasha, F. Llirbat, E. Simon, and P. Valduriez. Transaction chopping: Algorithms and performance studies. *ACM Trans. Database Syst.*, 20(3), 1995.

26 Y. Sovran, R. Power, M. K. Aguilera, and J. Li. Transactional storage for geo-replicated systems. In *SOSP*, 2011.

27 D. B. Terry, A. J. Demers, K. Petersen, M. Spreitzer, M. Theimer, and B. W. Welch. Session guarantees for weakly consistent replicated data. In *PDIS*, 1994.

28 P. Viotti and M. Vukolić. Consistency in non-transactional distributed storage systems. *ACM Comput. Surv.*, 49(1):19:1–19:34, June 2016.

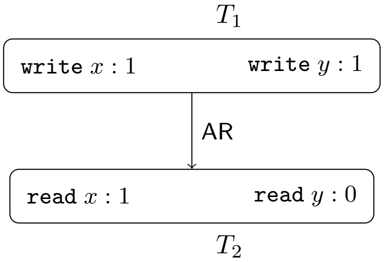
29 K. Zellag and B. Kemme. Consistency anomalies in multi-tier architectures: Automatic detection and prevention. *The VLDB Journal*, 23(1), 2014.

A Exampes of Anomalies

We give examples of several anomalies: for each of them we list those consistency models, among those considered in the paper, that allow the anomaly, and those that forbid it. For the sake of clarity, we have removed from the pictures below a transaction writing the initial value 0 to relevant objects, and visible to all other transactions. Also, unnecessary visibility and arbitration edges are omitted from the Figure.

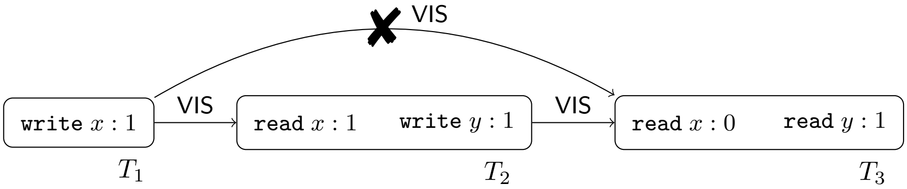
Fractured Reads: Transaction T_2 reads only one of the updates performed by transaction T_1:

- **Allowed by:** No consistency model enjoying atomic visibility allows this anomaly.



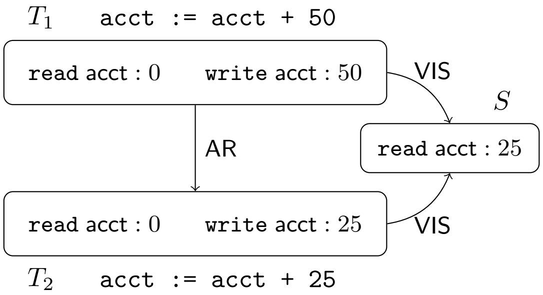
Violation of Causality: The update of transaction T_2 to object y depends on the value of x written by another transaction T_1. For example, T_2 can be generated by the code if$(x = 1)$ then $y := 1$;. A third transaction T_3 observes the update to y, but not the one to x.

- **Allowed by:** None of the models discussed in the paper. However, some other consistency models such as **Read Atomic** [4] allow this anomaly.



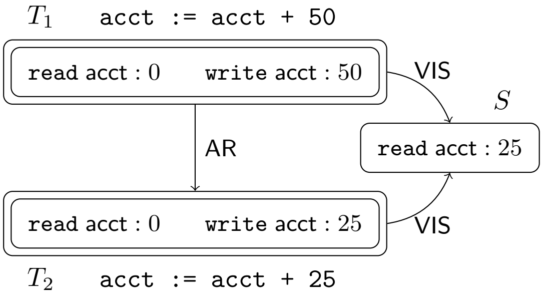
Lost Update: This is the abstract Execution depicted in Figure 1, which we draw again below. Two transactions T_1, T_2 concurrently update the state of the same object, after reading the initial value for it.

- **Allowed by:** Causal Consistency, Red-blue Consistency,
- **Forbidden by:** Parallel Snapshot Isolation, Snapshot Isolation, Serialisability.



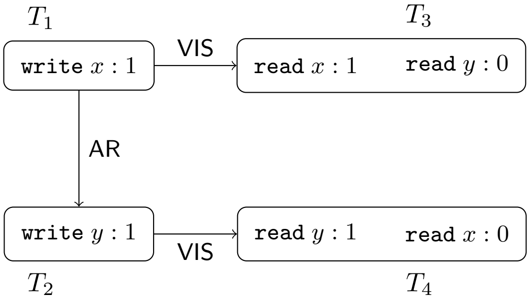
Serialisable Lost Update: This execution is the same as the one above, but the two transactions T_1, T_2 are marked as serialisable. This is represented in the figure below with a double box. Because Causal Consistency does not distinguish between transactions marked as serialisable from those that are not marked as such, it allows the serialisable lost update. However, this anomaly is forbidden by red-blue consistency.

- **Allowed by:** Causal Consistency,
- **Forbidden by:** Red-blue Consistency, Parallel Snapshot Isolation, Snapshot Isolation, Serialisability.



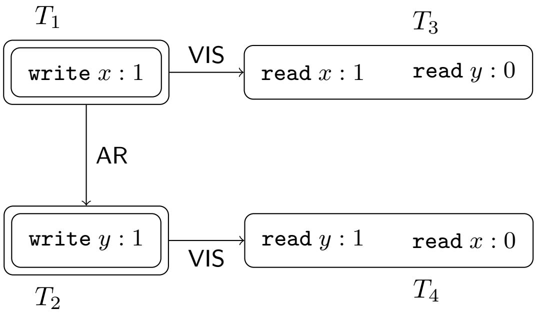
Long Fork: Two transactions T_1, T_2 write to different objects: two other transactions T_3, T_4 only observe the updates of T_1, T_2, respectively:

- **Allowed by:** Causal Consistency, Red-blue Consistency, Parallel Snapshot Isolation,
- **Forbidden by:** Snapshot Isolation, Serialisability.



Long Fork with Serialisable Updates: This is the same as the long fork, but the transactions T_1, T_2 that write to objects x, y, respectively, are marked as serialisable. Because Parallel Snapshot Isolation does not take serialisable transactions into account, it allows this anomaly. However, Red-Blue consistency distinguishes between serialisable and non-serialisable transactions, hence it does not allow it.

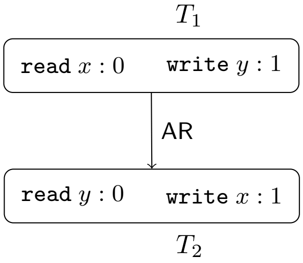
- **Allowed by:** Causal Consistency, Parallel Snapshot Isolation,
- **Forbidden by:** Red-blue Consistency, Snapshot Isolation, Serialisability.



Remark: Note that Red-blue consistency forbids this anomaly, but allows the lost update anomaly from above. In contrast, Parallel Snapshot Isolation allows this anomaly, but forbids the lost-update anomaly. In other words, Red-blue Consistency and Parallel Snapshot Isolation are incomparable: $\mathsf{Executions}(\Sigma_{\mathsf{RB}}) \nsubseteq \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$ and $\mathsf{Executions}(\Sigma_{\mathsf{PSI}}) \nsubseteq \mathsf{Executions}(\Sigma_{\mathsf{RB}})$.

Write Skew: Transactions T_1, T_2 read each the initial value of an object which is updated by the other.

- **Allowed by:** Causal Consistency, Red-blue Consistency, Parallel Snapshot Isolation, Snapshot Isolation,
- **Forbidden by:** Serialisability.



B Session Guarantees and Non-Causal Consistency Models

We augment histories with sessions: clients submit transactions within sessions, and the order in which they are submitted to the database is tracked by a *session order*. We propose a variant of x-specifications that allows for specifying session guarantees, as well as causality guarantees that are weaker than causal consistency.

▸ **Definition 23.** Let $\mathcal{T}$ be a set of transactions, and let $\{\mathcal{T}_1, \mathcal{T}_2, \cdots, \mathcal{T}_n\}$ be a partition of $\mathcal{T}$. An *extended history* is a pair $\mathcal{H} = (\mathcal{T}, \mathsf{SO})$, where $\mathsf{SO} = \bigcup_{i=1}^{n} \mathsf{SO}_i$, and each SO_i is a strict, total order over $\mathcal{T}_i$. Each of the sets $\mathcal{T}_i = 1, \cdots, n$ takes the name of *session*, and we call SO the *session order*.

Given a history $\mathcal{H} = (\mathcal{T}, \mathsf{SO})$, we let $\mathcal{T}_{\mathcal{H}} = \mathcal{T}$, and $\mathsf{SO}_{\mathcal{H}} = \mathsf{SO}$. If $(\mathcal{T}, \mathsf{SO})$ is a history, and $(\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ is an abstract execution, then we call $(\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ an *extended abstract execution*. Specification functions can also be lifted to take extended abstract executions into account: an *extended specification function* is a function $\rho : (\mathcal{H}, R) \mapsto R'$, such that for any extended history $\mathcal{H}$ and relation $R \subseteq \mathcal{T}_{\mathcal{H}} \times \mathcal{T}_{\mathcal{H}}$, $\rho(\mathcal{H}, R) = \rho(\mathcal{H}, \mathcal{T}_{\mathcal{H}} \times \mathcal{T}_{\mathcal{H}}) \cap R?$. An example of extended specification function is $\rho(\mathcal{H}, R) = R \backslash (\mathsf{SO}_{\mathcal{H}}?)$. An extended consistency guarantee is a pair (ρ, π), where ρ, π are extended specification functions.

▸ **Definition 24.** A *session guarantee* is a function $\sigma : 2^{\mathbb{T} \times \mathbb{T}} \to 2^{\mathbb{T} \times \mathbb{T}}$ such that, for any relation $R \subseteq \mathbb{T} \times \mathbb{T}$, $\sigma(R) \subseteq R?$. A *causality guarantee* is a pair (γ, β), where γ and β are extended specification functions.

An *extended x-specification* of a consistency model is a triple $\Sigma = (\{\sigma_i\}_{i\in I}, \{(\gamma_j, \beta_j)\}_{j\in J}, \{\rho_k,\ \pi_k\}_{k\in K})$, where I, J, K are (possibly empty) index sets, for any $i \in I, j_J$ and $k \in K$, σ_i is a session guarantee, (γ_j, β_j) is a causality guarantee, and (ρ_k, π_k) is an extended consistency guarantee.

Note that the definition of causality and (extended) consistency guarantees are the same. However, they play a different role when defining the set of executions admitted by a consistency model.

▸ **Definition 25.** An extended abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ conforms to the extended specification $(\{\sigma_i\}_{i\in I}, \{\gamma_j, \beta_j\}_{j\in J}, \{(\rho_k, \pi_k)\}_{k\in K}$ iff

1. for any $i \in I$, $\sigma_i(\mathsf{SO}) \subseteq \mathsf{VIS}$
2. for any $j \in J$, $\gamma_j(\mathcal{H}, \mathsf{VIS}) \ ; \beta_j(\mathcal{H}, \mathsf{VIS}) \subseteq \mathsf{VIS}$,
3. for any $k \in K$, $\rho_k(\mathcal{H}, \mathsf{VIS}) \ ; \mathsf{AR} \ ; \pi_k(\mathcal{H}, \mathsf{VIS}) \subseteq \mathsf{VIS}$.

Any x-specification can be lifted to an extended one: let $\gamma_{\mathsf{CC}}(\_, R) = (R\backslash\mathsf{Id})^5$. Let also Σ be any x-specification, and for any pair $(\rho, \pi) \in \Sigma$, define $\rho'(\_, R) = \rho(R)$, $\pi'(\_, R) = \pi(R)$. Then for any abstract $\mathcal{X}$, $\mathcal{X} \in \mathsf{Executions}(\Sigma)$ iff $\mathcal{X}$ conforms to the extended specification $(\emptyset, \{(\gamma_{\mathsf{CC}}, \gamma_{\mathsf{CC}}\}, \{(\rho', \pi') \mid (\rho, \pi) \in \Sigma\})$.

Dependency graphs can also be extended to take sessions into account. If $(\mathcal{T}, \mathsf{SO})$ is a history, and $(\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$ is a dependency graph, then $\mathcal{G} = (\mathcal{T}, \mathsf{SO}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$ is an *extended dependency graph*. Given an extended abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$, we define $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{SO}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$, where $(\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW}) = \mathsf{graph}(\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$. An extended abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ with underlying extended dependency graph $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{SO}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$ and conforming to the extended specification $(\{\sigma_i\}_{i\in I}, \{(\gamma_j, \beta_j)\}_{j\in J}, \{(\rho_k, \pi_k)\}_{k\in K}$, satisfies all the Equations of Figure 3, exception made for equations, **(c.8)** and **(c.9)**. Furthermore, sessions and causality guarantees induce novel inequations, which are listed below:

1. $\bigcup_{i\in I} \sigma_i(\mathsf{SO}) \subseteq \mathsf{VIS}$,
2. for any $j \in J$, $(\beta_j(\mathcal{H}, \mathsf{VIS}) \ ; \overline{\mathsf{VIS}^{-1}}) \cap \gamma(\mathcal{H}, \mathcal{T} \times \mathcal{T})^{-1} \subseteq \overline{\mathsf{VIS}^{-1}}$,
3. for any $j \in J$, $(\overline{\mathsf{VIS}^{-1}} \ ; \gamma_j(\mathcal{H}, \mathsf{VIS})) \cap \beta_j(\mathcal{H}, \mathcal{T} \times \mathcal{T})^{-1}\overline{\mathsf{VIS}^{-1}}$.

Equation (1) is obviously satisfied. To see why (3) is satisfied by $\mathcal{X}$, suppose that $T \xrightarrow{\beta(\mathcal{H},\mathsf{VIS})} V \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$, and $S \xrightarrow{\gamma(\mathcal{H},\mathcal{T}\times\mathcal{T})} T$. If it were $S \xrightarrow{\mathsf{VIS}} T$, then we would have a contradiction: because γ is a specification function, $S \xrightarrow{\gamma(\mathcal{H},\mathcal{T}\times\mathcal{T})} T$ and $S \xrightarrow{\mathsf{VIS}} T$ imply that $S \xrightarrow{\gamma(\mathcal{H},\mathsf{VIS})} T$, and together with $T \xrightarrow{\beta(\mathcal{H},\mathsf{VIS})} V$ then we would have $S \xrightarrow{\mathsf{VIS}} V$, contradicting the assumption that $V \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$. Therefore it has to be $\neg(S \xrightarrow{\mathsf{VIS}} T)$, or equivalently $T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$.

Examples of Session Guarantees:. Below we give some examples of session guarantees, inspired by [27].

Read Your Writes: This guarantee states that when processing a transaction, a client must see previous writes in the same session. This can be easily expressed via the collection of consistency guarantees $\{\sigma_{\mathsf{RYW(x)}}(R)\}_{x\in\mathsf{Obj}}$, where for each object x, $\sigma_{\mathsf{RYW(x)}}(R) = [\mathsf{Writes}_x] \ ; R \ ; [\mathsf{Reads}_x]$. An extended abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ satisfies this session guarantee if $\bigcup_{x\in\mathsf{Obj}} [\mathsf{Writes}_x] \ ; \mathsf{SO} \ ; [\mathsf{Reads}_x] \subseteq \mathsf{VIS}$,

Monotonic Writes: This guarantee states that transactions writing at least to one object are processed in the same order in which the client requested them. It can be specified via the function $\sigma_{\mathsf{MW}}(R) = (\bigcup_{x\in\mathsf{Obj}} [\mathsf{Writes}_x]) \ ; R \ ; (\bigcup_{x\in\mathsf{Obj}} [\mathsf{Writes}_x])$. Any extended abstract execution

[5] The difference with the identity relation is needed for γ to satisfy the definition of specification function. However, we will always apply γ to an irreflexive relation R, for which $\gamma(\_, R) = (R\backslash\mathsf{Id}) = R$.

$\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ satisfies the monotonic writes guarantee, is such that $(\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x])$; SO ; $(\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x]) \subseteq \mathsf{VIS}$,

Strong Session Guarantees: This guarantee states that all transactions are processed by the database in the same order in which the client requested them. It can be specified via the function $\sigma_{\mathsf{SS}}(R) = R$; an extended abstract execution $(\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ satisfies this guarantee if $\mathsf{SO} \subseteq \mathsf{VIS}$.

Examples of Causality Guarantee: . We have already seen how to model causal consistency via the causality guarantee $(\gamma_{\mathsf{CC}}, \gamma_{\mathsf{CC}})$. Below we give an example of weak causality guarantee:

Per-object Causal Consistency: this guarantee states that causality is preserved only among transactions accessing the same object. That is, let $\gamma_x(R) = ([\mathsf{Writes}_x \cup \mathsf{Reads}_x] \; ; R \; ; [\mathsf{Writes}_x \cup \mathsf{Reads}_x]) \setminus \mathsf{Id}$. The difference with the identity set is needed in order for $\gamma_x(R)$ to be a specification function. By definition, An extended abstract execution $\mathcal{X} = (\mathcal{T}, \mathsf{SO}, \mathsf{VIS}, \mathsf{AR})$ that satisfies the per-object causal consistency guarantee, satisfies the inequation $[\mathsf{Writes}_x \cup \mathsf{Reads}_x]$; VIS ; $[\mathsf{Writes}_x \cup \mathsf{Reads}_x]$; VIS ; $[\mathsf{Writes}_x \cup \mathsf{Reads}_x] \subseteq \mathsf{VIS}$.

C Additional Proofs of Algebraic Laws and Necessary Acyclicity Conditions

Throughout this Section, we assume that $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ is a valid abstract execution, and $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$.

First, a result about specification functions, which was hinted at in the main paper:

▸ Proposition 26. Let $\rho(\cdot)$ be a specification function. For all histories $\mathcal{T}$ and relations $R, R' \subseteq \mathcal{T} \times \mathcal{T}$,

(i) $\rho(R) \subseteq R?$;
(ii) $\rho(\mathcal{T} \times \mathcal{T}) \cap R \subseteq \rho(R)$;
(iii) $\rho(R) \cup \rho(R') = \rho(R \cup R')$.

Proof. Recall that, by definition, if ρ is a specification function, then $\rho(R) = \rho(\mathcal{T} \times \mathcal{T}) \cap R?$. It is immediate to observe then that (i) $\rho(R) \subseteq R?$, and (ii) $\rho(\mathcal{T} \times \mathcal{T}) \cap R \subseteq \rho(\mathcal{T} \times \mathcal{T}) \cap R? = \rho(R)$. To prove (iii) note that

$$\rho(R) \cup \rho(R') = (\rho(\mathcal{T} \times \mathcal{T}) \cap R?) \cup (\rho(\mathcal{T} \times \mathcal{T}) \cap R'?) = \rho(\mathcal{T} \times \mathcal{T}) \cap (R? \cup R'?) = \rho(\mathcal{T} \times \mathcal{T}) \cap (R \cup R')? = \rho(R \cup R')$$

◂

C.1 Proof of the Algebraic Laws in Figure 3

▸ Proposition 27. All the inequations in Figure 3**(a)** are satisfied.

Proof. We prove each of the Equations in Figure 3**(a)** individually. Throughout the proof, we let $\mathcal{T}', \mathcal{T}_1, \mathcal{T}_2 \subseteq \mathcal{T}$, and $R_1, R_2 \subseteq \mathcal{T} \times \mathcal{T}$

- **(a.1)**: by Definition, $[\mathcal{T}'] = \{(T, T) \mid T \in \mathcal{T}\} \subseteq \mathsf{Id}_{\mathcal{T}}$,
- **(a.2)**: note that we can rewrite $[\mathcal{T}_i] = \{(T, S) \mid T \in \mathcal{T}_1 \wedge S \in \mathcal{T}_1 \wedge T = S\}$, where $i = 1, 2$; then

$$\begin{aligned}
&[\mathcal{T}_1] \; ; [\mathcal{T}_2] = \{(T, S) \mid \exists V. \, (T, V) \in [\mathcal{T}_1] \wedge (V, S) \mid [\mathcal{T}_2]\} = \\
&\{(T, S) \mid \exists V. \, T \in \mathcal{T}_1 \wedge V \in \mathcal{T}_1 \wedge T = V \wedge S \in \mathcal{T}_2 \wedge V \in \mathcal{T}_2 \wedge V = S\} = \\
&\{(T, S) \mid T \in \mathcal{T}_1 \wedge S \in \mathcal{T}_1 \wedge S = V \wedge S \in \mathcal{T}_2 \wedge T \in \mathcal{T}_2\} = \\
&\qquad \{(T, S) \mid T \in (\mathcal{T}_1 \cap \mathcal{T}_2) \wedge S \in (\mathcal{T}_1 \cap \mathcal{T}_2) \wedge (S = T)\} = [\mathcal{T}_1 \cap \mathcal{T}_2]
\end{aligned}$$

- **(a.3):**

$$(R_1 ; [\mathcal{T}']) \cap R_2 = \{(T,S) \mid (\exists V.\,(T,V) \in R_1 \wedge V \in \mathcal{T}' \wedge V = S) \wedge (T,S) \in R_2\} = \{(T,S) \mid (T,S) \in R_1 \cap R_2 \wedge S \in \mathcal{T}'\} = (R_1 \cap R_2) ; [\mathcal{T}']$$

- **(a.4):**

$$([\mathcal{T}'] ; R_1) \cap R_2 = \{(T,S) \mid= (\exists V. T = V \wedge T \in \mathcal{T}' \wedge (V,S) \in R_1) \wedge (T,S) \in R_2\} = \{(T,S) \mid (T,S) \in R_1 \cap R_2 \wedge T \in \mathcal{T}'\} = [\mathcal{T}'] ; (R_1 \cap R_2)$$

◀

▸ Proposition 28. All the inequations of Figure 3**(b)** are satisfied by $\mathcal{X}$.

Proof. We only prove inequations **(b.1)** and **(b.4)**. The proof for the other equations is similar.

Suppose that $(T,S) \in \mathsf{WR}(x)$. By Definition, $T \ni \texttt{read}\ x : \_$, hence $(T,T) \in [\mathsf{Reads}_x]$. Also, $S \in \mathsf{VIS}^{-1} \cap \mathsf{Writes}_x \subseteq \mathsf{Writes}_x$, from which $(S,S) \in [\mathsf{Writes}_x]$ follows. Thus, $(T,S) \in [\mathsf{Reads}_x] ; \mathsf{WR}(x) ; [\mathsf{Writes}_x]$; this proves Equation **(b.1)**.

Also, because $T \in \mathsf{VIS}^{-1}(S)$, then $\mathsf{VIS} \subseteq \mathsf{AR}$ and $\mathsf{AR} \cap \mathsf{Id} \subseteq \varnothing$: by Definition of abstract execution, then $T \neq S$. Therefore, $\mathsf{WR}(x) \cap \mathsf{Id} = \varnothing$. Now we can rewrite

$$\mathsf{WR}(x) = (\mathsf{WR}(x) \cap (\mathsf{Id} \cup \overline{\mathsf{Id}})) = (\mathsf{WR}(x) \cap \mathsf{Id}) \cup \mathsf{WR}(x) \cap \overline{\mathsf{Id}} = \varnothing \cup (\mathsf{WR}(x) \cap \overline{\mathsf{Id}}) = \mathsf{WR}(x) \cap \overline{\mathsf{Id}} = \mathsf{WR}(x) \backslash \mathsf{Id}.$$

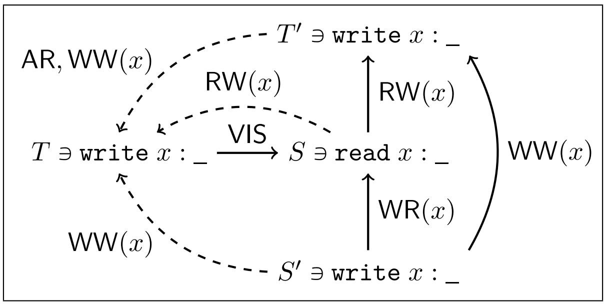
◀

▸ Proposition 29. $\mathcal{X}$ satisfies the inequation **(c.3)**.

Proof. Let T, S be such that $T \xrightarrow{\mathsf{RW}(x)} S$; by Definition, $T \neq S$, $T \ni \texttt{read}\ x : n$, $S \ni \texttt{write}\ x : m$ for some $n, m \in \mathbb{N}$. Also, there exists a $T' \in \mathcal{T}$ such that $T' \xrightarrow{\mathsf{WR}(x)} T, T' \xrightarrow{\mathsf{WW}(x)} S$. Because $T' \xrightarrow{\mathsf{WR}(x)} T$, by definition of $\mathsf{graph}(\mathcal{X})$ it follows that $T' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x)$. Because $T' \xrightarrow{\mathsf{WW}(x)} S$ and because of the inequation **(c.2)**, it must be the case that $T' \xrightarrow{\mathsf{AR}} S$. Because $T' \xrightarrow{\mathsf{AR}} S$, $S \in \mathsf{Writes}_x$, and $T' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x$, it must be the case that $\neg(S \xrightarrow{\mathsf{VIS}} T)$, or equivalently $T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$. ◀

▸ Proposition 30. $\mathcal{X}$ satisfies inequations **(c.1)**, **(c.2)** and **(c.7)**.

Proof. The inequations **(c.1)** and **(c.2)** follow directly from the Definition of $\mathsf{graph}(\mathcal{X})$. It remains to prove the inequation **(c.7)**. Let T, S, T' be three transactions such that $T \ni \texttt{write}\ x : \_$, $T \xrightarrow{\mathsf{VIS}} S$ and $S \xrightarrow{\mathsf{RW}(x)} T'$; we need to show that $T \xrightarrow{\mathsf{AR}} T'$. Recall that, because $\mathcal{X}$ is an abstract execution, then the relation AR is total: either $T = T'$, $T' \xrightarrow{\mathsf{AR}} T$, or $T \xrightarrow{\mathsf{AR}} T$. It is not possible that $T = T'$, because otherwise we would have $S \xrightarrow{\mathsf{RW}(x)} T$ and $T \xrightarrow{\mathsf{VIS}} S$ (equivalently, $\neg(S \xrightarrow{\overline{\mathsf{VIS}^{-1}}} T)$), contradicting Equation **(c.3)**. It cannot be that $T' \xrightarrow{\mathsf{AR}} T$ either: in the picture to the right, we have given a graphical representation of this scenario, where dashed edges represent the consequences of having $T' \xrightarrow{\mathsf{AR}} T$. In this case, $T \in \mathsf{Writes}_x$ by hypothesis; because $S \xrightarrow{\mathsf{RW}(x)} T'$, we also have that $T' \in \mathsf{Writes}_x$; because $T, T' \in \mathsf{Writes}_x$, and $T' \xrightarrow{\mathsf{AR}} T$, the definition of $\mathsf{graph}(\mathcal{X})$ implies that it has to be $T' \xrightarrow{\mathsf{WW}(x)} T$. Since $S \xrightarrow{\mathsf{RW}(x)} T$, then $S' \xrightarrow{\mathsf{WR}(x)} T$,

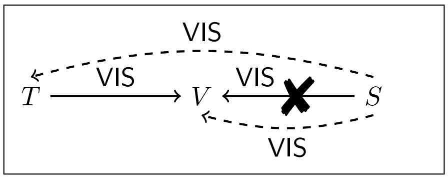

$T' \ni \texttt{write}\ x : \_$
AR, WW(x)
RW(x)
RW(x)
VIS
$T \ni \texttt{write}\ x : \_$
$S \ni \texttt{read}\ x : \_$
WW(x)
WR(x)
WW(x)
$S' \ni \texttt{write}\ x : \_$

and $S' \xrightarrow{\mathsf{WW}(x)} T$ for some S'; because $\mathsf{WW}(x)$ is transitive, then $S' \xrightarrow{\mathsf{WW}(x)} T'$. We have proved that $S' \xrightarrow{\mathsf{WR}(x)} S$, and $S' \xrightarrow{\mathsf{WW}(x)} T'$. By definition, it follows that $S \xrightarrow{\mathsf{RW}(x)} T$: together with the hypothesis $T \xrightarrow{\mathsf{VIS}} S$, we get a contradiction because the inequation **(c.3)** is violated. We have proved that it cannot be $T = T'$, nor $T' \xrightarrow{\mathsf{AR}} T$. Therefore $T \xrightarrow{\mathsf{AR}} T'$, as we wanted to prove. ◂

▸ Proposition 31. $\mathcal{X}$ satisfies inequations **(c.8)** and **(c.9)**.

Proof. We only prove the inequation **(c.8)**, as the inequation **(c.9)** can be proved in a similar manner.

Suppose that $T \xrightarrow{\mathsf{VIS}} V \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$. We prove that $\neg(S \xrightarrow{\mathsf{VIS}} T)$, or equivalently $(T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S)$, by contradiction. Let then $S \xrightarrow{\mathsf{VIS}} T$. Because $\mathcal{X}$ respects causality, $S \xrightarrow{\mathsf{VIS}} T \xrightarrow{\mathsf{VIS}} V$ implies that $S \xrightarrow{\mathsf{VIS}} V$. But $V \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$ by hypothesis, which causes the contradiction. A graphical representation of the proof is given to the right; here dashed edges are implied by the assumption that $S \xrightarrow{\mathsf{VIS}} T$. ◂



▸ Proposition 32. $\mathcal{X}$ satisfies all the inequations of Figure 3(**c**).

Proof. We have proved that $\mathcal{X}$ satisfies the inequations **(c.1)**, **(c.2)**, **(c.3)** and **(c.7)** in propositions 29 and 30. The inequations **(c.5)**, **(c.6)**, and **(c.12)** are trivial consequences of the definition of abstract execution. The inequations **(c.4)** is satisfied because we are assuming that $\mathcal{X}$ respects causality. The inequation **(c.11)** is a trivial consequence of the fact that, for any relation $R \subseteq \mathcal{T} \times \mathcal{T}$, $\overline{R^{-1}} = \{(T,S) \mid (S,T) \notin R\}$; then

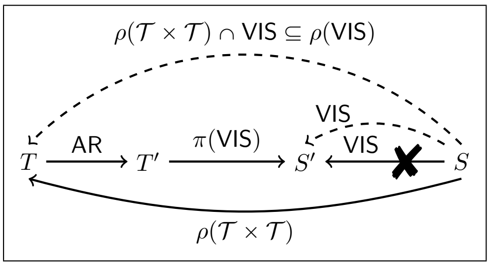
$$(R \,;\, \overline{R^{-1}}) \cap \mathsf{Id} = \{(T,T) \mid \exists S.\,(T,S) \in R \wedge (S,T) \in \overline{R^{-1}}\} = \\ \{(T,T) \mid \exists S.\,(T,S) \in R \wedge (T,S) \notin R\} = \emptyset$$

The inequation **(c.10)** can be proved similarly. Finally, the inequations **(c.8)** and **(c.9)** are satisfied, as we have proved in Proposition 31. ◂

▸ Proposition 33. If $\mathcal{X}$ satisfies the consistency guarantee (ρ, π), then it also satisfies the inequations **(d.3)** and **(d.4)**.

Proof. We only prove the inequation **(d.3)**. The proof for the inequaiton **(d.4)** is similar. Let $T, T', S', S \in \mathcal{T}$ be such that $T \xrightarrow{\mathsf{AR}} T'$, $T' \xrightarrow{\pi(\mathsf{VIS})} S'$, $S' \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$, and $S \xrightarrow{\rho(\mathcal{T}\times\mathcal{T})} T$.

We need to prove that $T \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$, or equivalently that $\neg(S \xrightarrow{\mathsf{VIS}} T)$. The proof goes by contradiction: suppose that $S \xrightarrow{\mathsf{VIS}} T$. Then we have that $S \xrightarrow{\rho(\mathcal{T}\times\mathcal{T})\cap\mathsf{VIS}} T$, and by Proposition 26 it follows that $S \xrightarrow{\rho(\mathsf{VIS})} T$. We have $S \xrightarrow{\rho(\mathsf{VIS})} T \xrightarrow{\mathsf{AR}} T' \xrightarrow{\pi(\mathsf{VIS})} S'$. Because $\mathcal{X} \in \mathsf{Executions}(\{\rho, \pi\})$, then $S \xrightarrow{\mathsf{VIS}} S'$ by Inequation **(d.1)**. But $S' \xrightarrow{\overline{\mathsf{VIS}^{-1}}} S$ by hypothesis, hence the contradiction. A graphical representation of the proof is given to the right: here dashed edges are implied by the assumption that $S \xrightarrow{\mathsf{VIS}} T$. ◂



▸ Proposition 34. If $\mathcal{X}$ satisfies the consistency guarantee (ρ, π), then it satisfies all the inequations of Figure 3(**d**), relatively to said consistency guarantee.

Proof. Because $\mathcal{X}$ satisfies the consistency guarantee (ρ, π) by hypothesis, then it satisfies the inequation **(d.1)**. It also satisfies the inequation **(d.2)**, as we showed in §4. Finally, it satisfies inequations **(d.3)** and **(d.4)** by Proposition 33. ◂

C.2 Additional Algebraic Laws

Here we prove some additional algebraic laws that can be proved from the laws of Figure 3, and from the axioms of the Kleene Algebra and boolean algebra of set relations. In the following, we assume that $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ is an abstract execution, and $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$.

▸ Proposition 35. For all relations $R_1, R_2 \subseteq \mathcal{T} \times \mathcal{T}$,

$$(R_1 \,;\, R_2) \cap \mathsf{Id} \subseteq \emptyset \implies (R_2 \,;\, R_1) \cap \mathsf{Id} \subseteq \emptyset \tag{1}$$

Proof. Suppose $(R_1 \,;\, R_2) \cap \mathsf{Id} \subseteq \emptyset$. For any $T \in \mathcal{T}$, there exists no $S \in \mathcal{T}$ such that $(T, S) \in R_1$ and $(S, T) \in R_2$. In particular, there exists no $S \in T$ such that $(S, T) \in R_2, (T, S) \in R_1$, for all $T \in \mathcal{T}$: equivalently, $(S, S) \notin (R_2 \,;\, R_1)$. That is, $(R_2 \,;\, R_1) \cap \mathsf{Id} \subseteq \emptyset$. ◂

▸ Proposition 36. For any set $\mathcal{T}' \subseteq \mathcal{T}$,

$$[\mathcal{T}'] = [\mathcal{T}'] \,;\, [\mathcal{T}']. \tag{2}$$

Proof. $[\mathcal{T}'] = [\mathcal{T}' \cap \mathcal{T}'] \overset{\textbf{(a.2)}}{=} [\mathcal{T}'] \,;\, [\mathcal{T}']$. ◂

▸ Proposition 37.

$$\mathsf{WR} \subseteq \mathsf{VIS} \tag{3}$$

$$\mathsf{WW} \subseteq \mathsf{AR} \tag{4}$$

$$\mathsf{RW} \subseteq \overline{\mathsf{VIS}^{-1}} \tag{5}$$

Proof. We only give details for Equation (3); the other equations can be proved similarly:

$$\mathsf{WR} = \bigcup_{x \in \mathsf{Obj}} \mathsf{WR}(x) \overset{\textbf{(c.1)}}{\subseteq} \bigcup_{x \in \mathsf{Obj}} \mathsf{VIS} = \mathsf{VIS}.$$

◂

▸ Proposition 38.

$$\mathsf{WR} \subseteq \mathsf{WR} \backslash \mathsf{Id} \tag{6}$$

$$\mathsf{WW} \subseteq \mathsf{WW} \backslash \mathsf{Id} \tag{7}$$

$$\mathsf{RW} \subseteq \mathsf{RW} \backslash \mathsf{Id} \tag{8}$$

Proof. We only give details for Equation (6). The other equations can be proved similarly.

$$\mathsf{WR} = \bigcup_{x \in \mathsf{Obj}} \mathsf{WR}(x) \overset{\textbf{(c.1)}}{\subseteq} \bigcup_{x \in \mathsf{Obj}} (\mathsf{WR}(x) \backslash \mathsf{Id}) = \left(\bigcup_{x \in \mathsf{Obj}} \mathsf{WR}(x) \right) \backslash \mathsf{Id} = \mathsf{WR} \backslash \mathsf{Id}.$$

◂

▸ Proposition 39. For any relation $R \subseteq \mathcal{T} \times \mathcal{T}$,

$$(R \cap \mathsf{Id} = \emptyset) \iff (R \subseteq R \backslash \mathsf{Id}). \tag{9}$$

Proof. Suppose $R \cap \mathsf{Id} = \emptyset$. Then

$$R = R \cap (\mathsf{Id} \cup \overline{\mathsf{Id}}) = (R \cap \mathsf{Id}) \cup (R \cap \overline{\mathsf{Id}}) = \emptyset \cup (R \backslash \mathsf{Id}) = (R \backslash \mathsf{Id}).$$

Now, suppose that $R \subseteq R \backslash \mathsf{Id}$. Then

$$(R \cap \mathsf{Id}) \subseteq (R \backslash \mathsf{Id}) \cap \mathsf{Id} = (R \cap \overline{\mathsf{Id}}) \cap \mathsf{Id} = R \cap (\overline{\mathsf{Id}} \cap \mathsf{Id}) = R \cap \emptyset = \emptyset.$$

◀

▸ **Corollary 40.**

$$\mathsf{VIS} \subseteq \mathsf{VIS} \backslash \mathsf{Id} \tag{10}$$

$$(\mathsf{VIS} \,;\, \overline{\mathsf{VIS}^{-1}}) \subseteq (\mathsf{VIS} \,;\, \overline{\mathsf{VIS}^{-1}}) \backslash \mathsf{Id} \tag{11}$$

$$(\overline{\mathsf{VIS}^{-1}} \,;\, \mathsf{VIS}) \subseteq (\overline{\mathsf{VIS}^{-1}} \,;\, \mathsf{VIS}) \backslash \mathsf{Id} \tag{12}$$

Proof. The inequation (11) is a trivial consequence of the inequations (9) and **(c.11)**. The inequation (12) is a trivial consequence of equations (9) and **(c.10)**. For the inequation (10), it suffices to prove that $\mathsf{VIS} \cap \mathsf{Id} = \emptyset$, then the result follows from Equation (9). But this is trivially true:

$$\mathsf{VIS} \cap \mathsf{Id} \overset{\textbf{(c.6)}}{\subseteq} \mathsf{AR} \cap \mathsf{Id} \overset{\textbf{(c.12)}}{\subseteq} \emptyset. \tag{13}$$

◀

▸ Proposition 41.

$$\mathsf{WR}(x) \subseteq [\mathsf{Writes}_x] \,;\, \mathsf{WR}(x) \tag{14}$$

$$\mathsf{WR}(x) \subseteq \mathsf{WR}(x) \,;\, [\mathsf{Reads}_x] \tag{15}$$

$$\mathsf{WW}(x) \subseteq [\mathsf{Writes}_x] \,;\, \mathsf{WW}(x) \tag{16}$$

$$\mathsf{WW}(x) \subseteq \mathsf{WW}(x) \,;\, [\mathsf{Writes}_x] \tag{17}$$

$$\mathsf{RW}(x) \subseteq [\mathsf{Reads}_x] \,;\, \mathsf{RW}(x) \tag{18}$$

$$\mathsf{RW}(x) \subseteq \mathsf{RW}(x) \,;\, [\mathsf{Writes}_x] \tag{19}$$

Proof. We only prove Equation (14); the proof for the other equations is similar.

$$\mathsf{WR}(x) \overset{\textbf{(b.1)}}{\subseteq} [\mathsf{Reads}_x] \,;\, \mathsf{WR}(x) \,;\, [\mathsf{Writes}_x] \overset{\textbf{(a.1)}}{\subseteq} \mathsf{Id} \,;\, \mathsf{WR}(x) \,;\, [\mathsf{Writes}_x] = \mathsf{WR}(x) \,;\, [\mathsf{Writes}_x]. \tag{20}$$

◀

▸ Proposition 42. Let Σ be a consistency guarantee such that $(\rho_x, \rho_x) \in \Sigma$, for some object $x \in \mathsf{Obj}$. If $\mathcal{X} \in \mathsf{Executions}(\Sigma)$, then

$$\mathsf{WW}(x) \subseteq \mathsf{VIS}. \tag{21}$$

Proof. By instantiating the inequation **(d.1)** for the consistency guarantee (ρ_x, ρ_x), we obtain that the following equation is valid for $\mathcal{X}$:

$$[\mathsf{Writes}_x] \,;\, \mathsf{AR} \,;\, [\mathsf{Writes}_x] \subseteq \mathsf{VIS}. \tag{22}$$

Therefore we have that

$$\mathsf{WW}(x) \overset{\textbf{(b.2)}}{\subseteq} [\mathsf{Writes}_x] \,;\, \mathsf{WW}(x) \,;\, [\mathsf{Writes}_x] \overset{\textbf{(c.2)}}{\subseteq} [\mathsf{Writes}_x] \,;\, \mathsf{AR} \,;\, [\mathsf{Writes}_x] \overset{(22)}{\subseteq} \mathsf{VIS}.$$

◀

▸ **Corollary 43.** *Let Σ be a consistency model such that $(\rho_x, \rho_x) \in \Sigma$ for any $x \in \mathsf{Obj}$. If $\mathcal{X} \in \mathsf{Executions}(\Sigma)$, then*

$$\mathsf{WW} \subseteq \mathsf{VIS}. \tag{23}$$

Proof. If $\mathcal{X} \in \mathsf{Executions}(\Sigma)$, then

$$\mathsf{WW} = \bigcup_{x \in \mathsf{Obj}} \mathsf{WW}(x) \overset{(21)}{\subseteq} \bigcup_{x \in \mathsf{Obj}} \mathsf{VIS} \subseteq \mathsf{VIS}.$$

◀

▸ **Corollary 44.** *Let Σ be a consistency model such that $(\rho_x, \rho_x) \in \Sigma$ for any $x \in \mathsf{Obj}$. If $\mathcal{X} \in \mathsf{Executions}(\Sigma)$, then*

$$(\mathsf{WR} \cup \mathsf{WW})^+ \subseteq \mathsf{VIS}. \tag{24}$$

Proof. If $\mathcal{X} \in \mathsf{Executions}(\Sigma)$, then

$$(\mathsf{WR} \cup \mathsf{WW})^+ \overset{(3),(23)}{\subseteq} \mathsf{VIS}^+ \overset{\textbf{(c.4)}}{\subseteq} \mathsf{VIS}.$$

◀

C.3 Proofs of Robustness Criteria

Proof of Theorem 12. First, observe that by instantiating inequations **(d.1)**, **(d.2)** to the consistency guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{Id}})$, we obtain:

$$\mathsf{AR} \subseteq \mathsf{VIS} \tag{25}$$

$$(\overline{\mathsf{VIS}^{-1}}) \setminus \mathsf{Id} \subseteq \mathsf{AR} \tag{26}$$

We need to prove that if $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SER}})$, then $(\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \cap \mathsf{Id} \subseteq \emptyset$. We show that the following inequations hold:

$$\mathsf{WR} \subseteq \mathsf{AR} \tag{27}$$

$$\mathsf{WW} \subseteq \mathsf{AR} \tag{28}$$

$$\mathsf{RW} \subseteq \mathsf{AR}. \tag{29}$$

From the three inequations above, it follows immediately that $(\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \cap \mathsf{Id} \subseteq \emptyset$, as we wanted to prove:

$$(\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \cap \mathsf{Id} \overset{(27,28,29)}{\subseteq} \mathsf{AR}^+ \cap \mathsf{Id} \overset{\textbf{(c.5)}}{\subseteq} \mathsf{AR} \cap \mathsf{Id} \overset{\textbf{(c.12)}}{\subseteq} \emptyset$$

It remains to prove inequations (27), (28) and (29).

Proof of inequation (27)**:** $\mathsf{WR} \overset{(3)}{\subseteq} \mathsf{VIS} \overset{\textbf{(c.6)}}{\subseteq} \mathsf{AR}$,

Proof of inequation (28)**:** $\mathsf{WW} \overset{(4)}{\subseteq} \mathsf{AR}$,

Proof of inequation (29)**:** $\mathsf{RW} \overset{(8)}{\subseteq} \mathsf{RW} \setminus \mathsf{Id} \overset{(5)}{\subseteq} \overline{\mathsf{VIS}^{-1}} \setminus \mathsf{Id} \overset{(26)}{\subseteq} \mathsf{AR}$.

◀

Proof of Theorem 13. Suppose that $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{SI}})$. Recall that $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}}) \in \Sigma_{\mathsf{SI}}$. By instantiating inequations **(d.1)** and **(d.2)** to this consistency guarantee, we obtain the following inequations:

$$(\mathsf{AR} \ ; (\mathsf{VIS} \setminus \mathsf{Id})) \subseteq \mathsf{VIS} \tag{30}$$

$$((\mathsf{VIS} \setminus \mathsf{Id}) \ ; \overline{\mathsf{VIS}^{-1}}) \setminus \mathsf{Id} \subseteq \mathsf{AR}. \tag{31}$$

In practice, the inequations above can be strengthened as follows:

$$\mathsf{AR} \mathbin{;} \mathsf{VIS} \subseteq \mathsf{VIS} \tag{32}$$

$$\mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}} \subseteq \mathsf{AR}. \tag{33}$$

Proof of inequation (32): $\mathsf{AR} \mathbin{;} \mathsf{VIS} \overset{(10)}{\subseteq} \mathsf{AR} \mathbin{;} (\mathsf{VIS}\backslash\mathsf{Id}) \overset{(30)}{\subseteq} \mathsf{VIS}$,
Proof of inequation (33):
$(\mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}}) \overset{(11)}{\subseteq} (\mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}})\backslash\mathsf{Id} \overset{(10)}{\subseteq} ((\mathsf{VIS}\backslash\mathsf{Id}) \mathbin{;} \overline{\mathsf{VIS}^{-1}})\backslash\mathsf{Id} \overset{(31)}{\subseteq} \mathsf{AR}$.

Also, because $(\rho_x, \rho_x) \in \mathsf{SI}$ for any object $x \in \mathsf{Obj}$, by Corollary 44 we have that inequation (24) is satisfied: $(\mathsf{WR} \cup \mathsf{WW})^+ \subseteq \mathsf{VIS}$.

We need to prove that $((\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} \mathsf{RW}?)^+ \cap \mathsf{Id} \subseteq \varnothing$. To this end, It suffices to prove the following inequations:

$$(\mathsf{WR} \cup \mathsf{WW}) \subseteq \mathsf{AR} \tag{34}$$

$$(\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} \mathsf{RW} \subseteq \mathsf{AR} \tag{35}$$

In fact, assuming that inequations (34) and (35) are satisfied, we obtain

$$((\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} \mathsf{RW}?)^+ \cap \mathsf{Id} = ((\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} (\mathsf{RW} \cup \mathsf{Id}))^+ \cap \mathsf{Id} =$$
$$((\mathsf{WR} \cup \mathsf{WW}) \cup ((\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} \mathsf{RW}))^+ \cap \mathsf{Id} \overset{(34,35)}{\subseteq} \mathsf{AR}^+ \cap \mathsf{Id} \overset{\textbf{(c.5)}}{\subseteq} \mathsf{AR} \cap \mathsf{Id} \overset{\textbf{(c.12)}}{\subseteq} \varnothing$$

where the second equality holds because the union distributed over the operator $\cdot \mathbin{;} \cdot$, and for any relation R, $R \mathbin{;} \mathsf{Id} = R$.

Proof of inequation (34): $(\mathsf{WR} \cup \mathsf{WW}) \overset{(3)}{\subseteq} (\mathsf{VIS} \cup \mathsf{WW}) \overset{\textbf{(c.6)},(4)}{\subseteq} (\mathsf{AR} \cup \mathsf{AR}) = \mathsf{AR}$,
Proof of inequation (35):
$(\mathsf{WR} \cup \mathsf{WW}) \mathbin{;} \mathsf{RW} \subseteq (\mathsf{WR} \cup \mathsf{WW})^+ \mathbin{;} \mathsf{RW} \overset{(24)}{\subseteq} \mathsf{VIS} \mathbin{;} \mathsf{RW} \overset{(5)}{\subseteq} \mathsf{VIS} \mathbin{;} \overline{\mathsf{VIS}^{-1}} \overset{(33)}{\subseteq} \mathsf{AR}$.

◀

Proof of Theorem 14. Recall that $\Sigma_{\mathsf{PSI}} = \{(\rho_x, \rho_x)\}_{x \in \mathsf{Obj}}$. By instantiating, for each object $x \in \mathsf{Obj}$, inequations **(d.1)** and **(d.2)**, we obtain that

$$[\mathsf{Writes}_x] \mathbin{;} \mathsf{AR} \mathbin{;} [\mathsf{Writes}_x] \subseteq \mathsf{VIS} \tag{36}$$

$$([\mathsf{Writes}_x] \mathbin{;} \overline{\mathsf{VIS}^{-1}} \mathbin{;} [\mathsf{Writes}_x])\backslash\mathsf{Id} \subseteq \mathsf{AR} \tag{37}$$

We need to prove that, for any $x \in \mathsf{Obj}$, $((\mathsf{WR} \cup \mathsf{WW})^* \mathbin{;} \mathsf{RW}(x))^+ \cap \mathsf{Id} \subseteq \varnothing$. To this end, it suffices that for any $x \in \mathsf{Obj}$, the following inequation is satisfied:

$$([\mathsf{Writes}_x] \mathbin{;} (\mathsf{WR} \cup \mathsf{WW})^* \mathbin{;} \mathsf{RW}(x))^+ \cap \mathsf{Id} \subseteq \varnothing. \tag{38}$$

In fact, assuming that Inequation (38) is satisfied, we can apply the following theorem of Kleene Algebra:

$$\forall R_1, R_2 \subseteq \mathcal{T} \times \mathcal{T}.(R_1 \mathbin{;} R_2)^+ = (R_1 \mathbin{;} (R_2 \mathbin{;} R_1)^* \mathbin{;} R_2) \tag{39}$$

By applying the equation (39) to the equation (38), where $R_1 = [\mathsf{Writes}_x]$ and $R_2 = (\mathsf{WR} \cup \mathsf{WW})^* \mathbin{;} \mathsf{RW}(x)$, we obtain the following:

$$\left([\mathsf{Writes}_x] \mathbin{;} \left(((\mathsf{WR} \cup \mathsf{WW})^* \mathbin{;} \mathsf{RW}(x)) \mathbin{;} [\mathsf{Writes}_x]\right)^* \mathbin{;} \left((\mathsf{WR} \cup \mathsf{WW})^* \mathbin{;} \mathsf{RW}(x)\right)\right) \cap \mathsf{Id} \subseteq \varnothing.$$

$$\tag{40}$$

Next, we apply Proposition 35, relatively to $R_1 = [\mathsf{Writes}_x]$, $R_2 = (\mathsf{WR} \cup \mathsf{WW})^* \;;\; \mathsf{RW}(x)$. We obtain the following:

$$\left(\left(((\mathsf{WR} \cup \mathsf{WW})^* \;;\; \mathsf{RW}(x)) \;;\; [\mathsf{Writes}_x]\right)^* \;;\; \left((\mathsf{WR} \cup \mathsf{WW})^* \;;\; \mathsf{RW}(x)\right) \;;\; [\mathsf{Writes}_x]\right) \cap \mathsf{Id} \subseteq \varnothing. \tag{41}$$

or, equivalently,

$$\left(((\mathsf{WR} \cup \mathsf{WW}^*) \;;\; \mathsf{RW}(x)) \;;\; [\mathsf{Writes}_x]\right)^+ \cap \mathsf{Id} = \varnothing. \tag{42}$$

By applying the inequation (19) - $\mathsf{RW}(x) = \mathsf{RW}(x) \;;\; [\mathsf{Writes}_x]$ - in inequation (42) above, we obtain the desired result:

$$((\mathsf{WR} \cup \mathsf{WW})^* \;;\; \mathsf{RW}(x))^+ \cap \mathsf{Id} \subseteq \varnothing. \tag{43}$$

It remains to prove Inequation (38). To this end, let $x \in \mathsf{Obj}$: we show that the following three inequations are satisfied:

$$[\mathsf{Writes}_x] \;;\; \mathsf{RW}(x) \subseteq \mathsf{AR} \tag{44}$$

$$[\mathsf{Writes}_x] \;;\; (\mathsf{WR} \cup \mathsf{WW})^+ \;;\; \mathsf{RW}(x) \subseteq \mathsf{AR}. \tag{45}$$

Assuming that (44) and (45) are satisfied, we obtain a proof for the inequation (38):

$$\begin{aligned}&([\mathsf{Writes}_x] \;;\; (\mathsf{WR} \cup \mathsf{WW})^* \;;\; \mathsf{RW}(x))^+ \cap \mathsf{Id} = \\ &\quad \left(([\mathsf{Writes}_x] \;;\; \mathsf{RW}(x)) \cup ([\mathsf{Writes}_x] \;;\; (\mathsf{WR} \cup \mathsf{WW})^+ \;;\; \mathsf{RW}(x))\right)^+ \overset{(44,45)}{\subseteq} \mathsf{AR}^+ \cap \mathsf{Id} \subseteq \varnothing.\end{aligned}$$

Proof of Inequation (44):

$$\begin{aligned}&[\mathsf{Writes}_x] \;;\; \mathsf{RW}(x) \overset{(8)}{\subseteq} [\mathsf{Writes}_x] \;;\; (\mathsf{RW}(x) \backslash \mathsf{Id}) \overset{\textbf{(a.4)}}{=} ([\mathsf{Writes}_x] \;;\; \mathsf{RW}(x)) \backslash \mathsf{Id} \overset{(19)}{\subseteq} \\ &\quad ([\mathsf{Writes}_x] \;;\; \mathsf{RW}(x) \;;\; [\mathsf{Writes}_x]) \backslash \mathsf{Id} \overset{\textbf{(c.3)}}{\subseteq} ([\mathsf{Writes}_x] \;;\; \overline{\mathsf{VIS}^{-1}} \;;\; [\mathsf{Writes}_x]) \backslash \mathsf{Id} \overset{(37)}{\subseteq} \mathsf{AR},\end{aligned}$$

Proof of Inequation (45):

$$[\mathsf{Writes}_x] \;;\; (\mathsf{WR} \cup \mathsf{WW})^+ \;;\; \mathsf{RW}(x) \overset{(24)}{\subseteq} [\mathsf{Writes}_x] \;;\; \mathsf{VIS} \;;\; \mathsf{RW}(x) \overset{\textbf{(c.7)}}{\subseteq} \mathsf{AR}.$$

◀

Proof of Theorem 15. Recall that $\Sigma_{\mathsf{RB}} = \{(\rho_S, \rho_S)\}$. By instantiating inequations **(d.1)** and **(d.2)**, relatively to the consistency guarantee (ρ_S, ρ_S), we have that the following inequalities are satisfied:

$$([\mathtt{SerTx}] \;;\; \mathsf{AR} \;;\; [\mathtt{SerTx}]) \subseteq \mathsf{VIS} \tag{46}$$

$$([\mathtt{SerTx}] \;;\; \overline{\mathsf{VIS}^{-1}} \;;\; [\mathtt{SerTx}]) \backslash \mathsf{Id} \subseteq \mathsf{AR}. \tag{47}$$

We need to prove that $(\mathsf{WR} \cup \mathsf{WW} \cup \Vdash\!\mathsf{RW}\!\dashv\!\Vert)^+ \cap \mathsf{Id} \subseteq \varnothing$, where we recall that $\Vdash\!\mathsf{RW}\!\dashv\!\Vert = [\mathtt{SerTx}] \;;\; \mathsf{WR}^* \;;\; \mathsf{RW} \;;\; \mathsf{WR}^* \;;\; [\mathtt{SerTx}]$. To this end, it suffices to show that

$$\mathsf{WR}^* \;;\; \mathsf{RW} \;;\; \mathsf{WR}^* \subseteq \overline{\mathsf{VIS}^{-1}} \backslash \mathsf{Id} \tag{48}$$

In fact, assuming that inequation (48) holds, we obtain that:

$$\Vdash\!\mathsf{RW}\!\dashv\!\Vert \subseteq \mathsf{AR}. \tag{49}$$

Proof of inequation (49):

$$\Vdash\mathsf{RW}\dashv\!\vert = [\texttt{SerTx}] ; \mathsf{WR}^* ; \mathsf{RW} ; \mathsf{WR}^* ; [\texttt{SerTx}] \overset{(48)}{\subseteq} [\texttt{SerTx}] ; (\overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}) ; [\texttt{SerTx}] \overset{\textbf{(a.3)}}{=} [\texttt{SerTx}] ; (\overline{\mathsf{VIS}^{-1}} ; [\texttt{SerTx}])\backslash\mathsf{Id} \overset{\textbf{(a.4)}}{=} ([\texttt{SerTx}] ; \overline{\mathsf{VIS}^{-1}} ; [\texttt{SerTx}])\backslash\mathsf{Id} \overset{(47)}{\subseteq} \mathsf{AR}$$

Finally, we can prove the desired result:

$$(\mathsf{WR} \cup \mathsf{WW} \cup \Vdash\mathsf{RW}\dashv\!\vert)^+ \cap \mathsf{Id} \overset{(3)}{\subseteq} (\mathsf{VIS} \cup \mathsf{WW} \cup \Vdash\mathsf{RW}\dashv\!\vert)^+ \cap \mathsf{Id} \overset{\textbf{(c.6)},(4),(49)}{\subseteq} \mathsf{AR}^+ \cap \mathsf{Id} \overset{\textbf{(c.5)}}{\subseteq} \mathsf{AR} \cap \mathsf{Id} \overset{\textbf{(c.12)}}{\subseteq} \varnothing.$$

◂

It remains to prove the inequation (48). To this end, we prove the following four inequations:

$$\mathsf{RW} \subseteq \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id} \tag{50}$$

$$\mathsf{WR}^+ ; \mathsf{RW} \subseteq \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id} \tag{51}$$

$$\mathsf{RW} ; \mathsf{WR}^+ \subseteq \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id} \tag{52}$$

$$\mathsf{WR}^+ ; \mathsf{RW} ; \mathsf{WR}^+ \subseteq \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}. \tag{53}$$

In fact, assuming that the four inequations above hold, we have that

$$\mathsf{WR}^* ; \mathsf{RW} ; \mathsf{WR}^* = \mathsf{RW} \cup (\mathsf{WR}^+ ; \mathsf{RW}) \cup (\mathsf{RW} ; \mathsf{WR}^+) \cup (\mathsf{WR}^+ ; \mathsf{RW} ; \mathsf{WR}^+) \subseteq \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}$$

Proof of Inequation (50): $\mathsf{RW} \overset{(8)}{\subseteq} \mathsf{RW}\backslash\mathsf{Id} \overset{(5)}{\subseteq} \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}$,
Proof of Inequation (51):

$$(\mathsf{WR}^+ ; \mathsf{RW}) \overset{(3)}{\subseteq} \mathsf{VIS}^+ ; \mathsf{RW} \overset{\textbf{(c.4)}}{\subseteq} \mathsf{VIS} ; \mathsf{RW} \overset{(5)}{\subseteq} \mathsf{VIS} ; \overline{\mathsf{VIS}^{-1}} \overset{(11)}{\subseteq} (\mathsf{VIS} ; \overline{\mathsf{VIS}^{-1}})\backslash\mathsf{Id} \overset{\textbf{(c.8)}}{\subseteq} \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}$$

Proof of Inequation (52):

$$(\mathsf{RW} ; \mathsf{WR}^+) \overset{(3)}{\subseteq} \mathsf{RW} ; \mathsf{VIS}^+ \overset{\textbf{(c.4)}}{\subseteq} \mathsf{RW} ; \mathsf{VIS} \overset{(5)}{\subseteq} \overline{\mathsf{VIS}^{-1}} ; \mathsf{VIS} \overset{(12)}{\subseteq} (\overline{\mathsf{VIS}^{-1}} ; \mathsf{VIS})\backslash\mathsf{Id} \overset{\textbf{(c.9)}}{\subseteq} \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}$$

Proof of Inequation (53):

$$(\mathsf{WR}^+ ; \mathsf{RW} ; \mathsf{WR}^+) \overset{(3)}{\subseteq} \mathsf{VIS}^+ ; \mathsf{RW} ; \mathsf{VIS}^+ \overset{\textbf{(c.4)}}{\subseteq} \mathsf{VIS} ; \mathsf{RW} ; \mathsf{VIS} \overset{(5)}{\subseteq} (\mathsf{VIS} ; \overline{\mathsf{VIS}^{-1}} ; \mathsf{VIS}) \overset{\textbf{(c.8)}}{\subseteq} \overline{\mathsf{VIS}^{-1}} ; \mathsf{VIS} \overset{(12)}{\subseteq} (\overline{\mathsf{VIS}^{-1}} ; \mathsf{VIS})\backslash\mathsf{Id} \overset{\textbf{(c.9)}}{\subseteq} \overline{\mathsf{VIS}^{-1}}\backslash\mathsf{Id}$$

◂

C.4 Additional Robustness Criteria

So far, none of the robustness criteria that we have derived has exploited the inequations **(d.3)** and **(d.4)** from Figure 3. Here we give another example of x-specification, for which we can derive a robustness criterion which makes use of the inequations **(d.3)** and **(d.4)**. Such a specification is given by $\Sigma = \{(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}}), (\rho_S, \rho_S)\}$. This can be thought as a weakening of $\Sigma_{\mathsf{SI+SER}}$ which does not have any write conflict detection. By applying inequations **(d.1)**, **(d.2)** to both consistency guarantees, we obtain the following:

$$\mathsf{AR} ; \mathsf{VIS} \subseteq \mathsf{VIS} \tag{54}$$

$$[\mathtt{SerTx}] ; \mathsf{AR} ; [\mathtt{SerTx}] \subseteq \mathsf{VIS} \tag{55}$$

$$\mathsf{VIS} ; \overline{\mathsf{VIS}^{-1}} \subseteq \mathsf{AR} \tag{56}$$

$$([\mathtt{SerTx}] ; \overline{\mathsf{VIS}^{-1}} ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \subseteq \mathsf{AR} \tag{57}$$

Also, we can instantiate inequation **(d.4)** relatively to the consistency guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$. Recall that $\rho_{\mathsf{Id}}(\mathsf{VIS}) = \mathsf{Id}$, and $\rho_{\mathsf{SI}}(\mathcal{T} \times \mathcal{T}) = \rho_{\mathsf{SI}}(\mathcal{T} \times \mathcal{T})^{-1} = (\mathcal{T} \times \mathcal{T}) \setminus \mathsf{Id}$. We have that $(\overline{\mathsf{VIS}^{-1}} ; \rho_{\mathsf{Id}}(\mathsf{VIS}) ; \mathsf{AR}) ; \rho_{\mathsf{SI}}(\mathcal{T} \times \mathcal{T})^{-1} = (\overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}) \cap ((\mathcal{T} \times \mathcal{T}) \setminus \mathsf{Id})^{-1} = (\overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}) \setminus \mathsf{Id}$. Therefore, we have the following:

$$(\overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}) \setminus \mathsf{Id} \subseteq \overline{\mathsf{VIS}^{-1}} \tag{58}$$

Using this inequations, we can derive a robustness criterion for the consistency model induced by the x-specification Σ.

▸ **Theorem 45.** *Let* $\mathcal{X} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR}) \in \mathsf{Executions}(\Sigma)$*, where* $\Sigma = \{(\rho_S, \rho_S), (\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})\}$*. We say that a path* $T_0 \xrightarrow{R_0} \cdots \xrightarrow{R_{n-1}} T_n$ *of* $\mathsf{graph}(\mathcal{X})$*, is* critical *if* $T_0 \neq T_n$*, both* $T_0, T_n \ni \mathtt{SerTx}$*, only one of the edges* $R_i, 0 \leqslant i < n$ *is an anti-dependency, and none of the edges* $R_j, 0 \leqslant j < i$ *is a* WW*-edge. Then all cycles of* $\mathsf{graph}(\mathcal{X})$ *have at least one anti-dependency edge that is not contained within a critical sub-path of the cycle.*

Formally, let $\mathsf{CSub} = ([\mathtt{SerTx}] ; \mathsf{WR}^* ; \mathsf{RW} ; (\mathsf{WW} \cup \mathsf{WR})^* ; [\mathtt{SerTx}]) \setminus \mathsf{Id}$*, where* $\mathsf{graph}(\mathcal{X}) = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$*. Then* $(\mathsf{WR} \cup \mathsf{WW} \cup ([\mathtt{SerTx}] ; \mathsf{CSub} ; [\mathtt{SerTx}]))$ *is acyclic.*

Proof Sketch. It suffices to prove the following:

$$\mathsf{CSub} \subseteq \mathsf{AR} \tag{59}$$

To see why this inequation (59) is satisfies, note that we have the following:

$$\mathsf{CSub} = ([\mathtt{SerTx}] ; \mathsf{WR}^* ; \mathsf{RW} ; (\mathsf{WW} \cup \mathsf{WR})^* ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{(3)}{\subseteq}$$

$$([\mathtt{SerTx}] ; \mathsf{VIS}^* ; \mathsf{RW} ; (\mathsf{WW} \cup \mathsf{VIS})^* ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{(4)}{\subseteq}$$

$$([\mathtt{SerTx}] ; \mathsf{VIS}^* ; \mathsf{RW} ; (\mathsf{AR} \cup \mathsf{VIS})^* ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{\textbf{(c.6)}}{\subseteq}$$

$$([\mathtt{SerTx}] ; \mathsf{VIS}^* ; \mathsf{RW} ; \mathsf{AR}^* ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{\textbf{(c.4)},\textbf{(c.5)}}{\subseteq}$$

$$([\mathtt{SerTx}] ; \mathsf{VIS}? ; \mathsf{RW} ; \mathsf{AR}? ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{(5)}{\subseteq}$$

$$([\mathtt{SerTx}] ; \mathsf{VIS}? ; \overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}? ; [\mathtt{SerTx})] \setminus \mathsf{Id} \overset{\textbf{(c.8)}}{\subseteq}$$

$$([\mathtt{SerTx}] ; \overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}? ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \subseteq$$

$$\Big(([\mathtt{SerTx}] ; \overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}? ; [\mathtt{SerTx}]) \setminus \mathsf{Id}\Big) \setminus \mathsf{Id} \overset{\textbf{(a.4)},\textbf{(a.3)}}{\subseteq}$$

$$\Big([\mathtt{SerTx}] ; (\overline{\mathsf{VIS}^{-1}} ; \mathsf{AR}?) \setminus \mathsf{Id} ; [\mathtt{SerTx}]\Big) \setminus \mathsf{Id} \overset{(58)}{\subseteq}$$

$$([\mathtt{SerTx}] ; \overline{\mathsf{VIS}^{-1}} ; [\mathtt{SerTx}]) \setminus \mathsf{Id} \overset{(57)}{\subseteq} \mathsf{AR}$$

D Proofs of Results for Simple x-Specifications

Let $X \subseteq \mathsf{Obj}$ and suppose that (ρ, π) is a consistency guarantee throughout this section we will work with the (simple) x-specification $\Sigma = \{(\rho_x, \rho_x)\}_{x \in X} \cup \{(\rho, \pi)\}$, although all the results apply to the x-specification $\Sigma' = \{(\rho_x, \rho_x)\}_{x \in X}$ which does not contain any consistency guarantee, aside from those enforcing the write conflict detection over some objects.

D.1 Proof of Proposition 20

Let $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$ be a dependency graph.

Recall the following definition of valid pre-execution:

▸ **Definition 46.** a pre-execution is a quadruple $\mathcal{P} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ such that

1. $\mathsf{VIS} \subseteq \mathsf{AR}$,
2. VIS and AR are strict partial orders,
3. for any object $x \in \mathsf{Obj}$, AR is total over the set Writes_x,
4. $\mathcal{P}$ satisfies the Last Write Wins property: for any $T \in \mathcal{T}$, if $T \ni \mathtt{read}\ x : n$ then $S := \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x)$ is well defined, and $S \ni \mathtt{write}\ x : n$.

The proof of Proposition 20 relies on the following auxiliary result:

▸ Proposition 47. Let $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ be a solution of $\mathsf{System}_\Sigma(\mathcal{G})$. If $\mathsf{AR} \cap \mathsf{Id}$ is acyclic, then $\mathcal{P} = (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$ is a valid pre-execution.

Proof. Because $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_\Sigma(\mathcal{G})$, all the inequalities in the latter are satisfied when substituting the relations $\mathsf{VIS}, \mathsf{AR}, \mathsf{AntiVIS}$ for the unknowns X_V, X_A, X_N, respectively. In particular, we have that and $\mathsf{VIS} \subseteq \mathsf{AR}$ because of the inequation (A2). Because AR is irreflexive by hypothesis, it also implies that $\mathsf{VIS} \cap \mathsf{Id} \subseteq \mathsf{AR} \cap \mathsf{Id} \subseteq \emptyset$. Together with the inequation (V2) it ensures that VIS is a strict, partial order. Similarly, the assumption that $\mathsf{AR} \cap \mathsf{Id}$ and the inequation (A4) imply that AR is a strict, partial order.

Next, we prove that for any transactions $T', T'' \in \mathcal{T}$ such that $T' \in \mathsf{Writes}_x, T'' \in \mathsf{Writes}_x$ and $T' \xrightarrow{\mathsf{AR}} T''$, it must be $T' \xrightarrow{\mathsf{WW}(x)} T''$. In fact, for such transactions we have that $T' \xrightarrow{\mathsf{AR}} T''$ implies that $T' \neq T''$, since we are assuming that $\mathsf{AR} \cap \mathsf{Id} = \emptyset$. By the same hypothesis and the fact that $\mathsf{AR}^+ \subseteq \mathsf{AR}$ (as a consequence of Equation (A4)), we obtain that $\neg(T'' \xrightarrow{\mathsf{AR}} T')$. Because of the inequation (A1), $\mathsf{WW}(x) \subseteq \mathsf{WW} \subseteq \mathsf{AR}$, from which it follows that that $\neg(T'' \xrightarrow{\mathsf{WW}(x)} T')$. But $T' \in \mathsf{Writes}_x, T'' \in \mathsf{Writes}_x$, and because $\mathsf{WW}(x)$ is a total order over Writes_x, and $T' \neq T''$, it follows that the only possibility left is that $T' \xrightarrow{\mathsf{WW}(x)} T''$.

We have proved that for any two transactions T', T'' such that $T' \in \mathsf{Writes}_x, T'' \in \mathsf{Writes}_x$, $T' \xrightarrow{\mathsf{AR}} T''$ implies $T' \xrightarrow{\mathsf{WW}(x)} T''$. By the inequation (A1), this implication can be strengthened to an if and only if condition: the relation AR, restricted to transactions in the set Writes_x, coincides with $\mathsf{WW}(x)$. A trivial consequence of this fact is that AR is a strict, total order over Writes_x.

It remains to show that $\mathcal{P}$ satisfies the Last Write Wins property: to this end, let $T \in \mathcal{T}$ be a transaction such that $T \ni \mathtt{read}\ x : n$. By Definition 7 there exists a transaction S such that $S \ni \mathtt{write}\ x : n$ and $S \xrightarrow{\mathsf{WR}(x)} T$. By Equation (V1), we have that $\mathsf{WR} \subseteq \mathsf{VIS}$, hence $S \xrightarrow{\mathsf{VIS}} T$. Because $S \xrightarrow{\mathsf{VIS}} T$ and $S \ni \mathtt{write}\ x : n$, we have that $S \in (\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x)$, and in particular $(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x) \neq \emptyset$.

A consequence of the two facts above - $(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x) \neq \emptyset$, and $\mathsf{AR} \cap (\mathsf{Writes}_x \times \mathsf{Writes}_x) = \mathsf{WW}(x)$ - is that the entity $S' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x)$ is well-defined. It remains to prove that $S' \ni \mathtt{write}\ x : n$. To this end, it suffices to show that $S = S'$ (recall that S is the unique transaction such that $S \xrightarrow{\mathsf{WR}(x)} T$), and observe that $S \ni \mathtt{write}\ x : n$, from which the claim follows. Because $S, S' \in \mathsf{Writes}_x$ and $\mathsf{WW}(x)$ coincides with the restriction of AR to the set Writes_x, we obtain that either $S' \xrightarrow{\mathsf{AR}} S$, $S \xrightarrow{\mathsf{AR}} S'$ or $S = S'$. The first case is not possible, because $S \in \mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x$, and $S' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(T) \cap \mathsf{Writes}_x)$. The second case is also not possible: if $S \xrightarrow{\mathsf{AR}} S'$ then $S \xrightarrow{\mathsf{WW}(x)} S'$; together with $S \xrightarrow{\mathsf{WR}(x)} T$ this implies that there is an anti-dependency edge $T \xrightarrow{\mathsf{RW}(x)} S'$; now we have that $S' \in \mathsf{Writes}_x$, and $S' \xrightarrow{\mathsf{VIS}} T \xrightarrow{\mathsf{RW}(x)} S'$: that is, $(S', S') \in [\mathsf{Writes}_x] \ ; \ \mathsf{VIS} \ ; \ \mathsf{WR}(x)$. By the inequation (A3), this implies that $S' \xrightarrow{\mathsf{AR}} S'$, contradicting the assumption that $\mathsf{AR} \cap \mathsf{Id} \subseteq \emptyset$. We are left with the only possibility $S = S'$, which is exactly what we wanted to prove. ◀

Proof of Proposition 20. Let $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ be a solution to $\mathsf{System}_{\Sigma}(\mathcal{G})$, and suppose that Let $\mathcal{P} := (\mathcal{T}, \mathsf{VIS}, \mathsf{AR})$. By Proposition 47 we know that $\mathcal{P}$ is a valid pre-execution. To show that $\mathcal{P} \in \mathsf{PreExecutions}(\Sigma)$, we need to show the following:

1. for any object $x \in X$, $[\mathsf{Writes}_x] \ ; \ \mathsf{AR} \ ; \ [\mathsf{Writes}_x] \subseteq \mathsf{VIS}$; this is because we are assuming that $(\rho_x, \rho_x) \in \Sigma$ for any $x \in X$. Let then $x \in X$, and consider two transactions T, S be such that $T \xrightarrow{\mathsf{AR}} S$, and $T, S \in \mathsf{Writes}_x$: we show that $T \xrightarrow{\mathsf{VIS}} S$. Because $\mathsf{AR} \cap \mathsf{Id} \subseteq \emptyset$, then $T \neq S$. Also, it cannot be $S \xrightarrow{\mathsf{WW}(x)} T$: by inequation (A1) this would imply that $S \xrightarrow{\mathsf{AR}} T$; by inequation (A4) and the assumption that $T \xrightarrow{\mathsf{AR}} S$, this would lead to $S \xrightarrow{\mathsf{AR}} S$, contradicting the assumption that $\mathsf{AR} \cap \mathsf{Id} = \emptyset$. We have proved that $T, S \in \mathsf{Writes}_x$, $T \neq S$ and $\neg(S \xrightarrow{\mathsf{WW}(x)} T)$: since $\mathsf{WW}(x)$ is a total order over the set Writes_x, it must be $T \xrightarrow{\mathsf{WW}(x)} S$. It follows from the inequation (V3) that $T \xrightarrow{\mathsf{VIS}} S$,
2. $\rho(\mathsf{VIS}) \ ; \ \mathsf{AR} \ ; \ \pi(\mathsf{VIS}) \subseteq \mathsf{VIS}$; this inequality is directly enforced by the inequation (V4).

Therefore, $\mathcal{P}$ is a valid pre-execution such that $[\mathsf{Writes}_x] \ ; \ \mathsf{AR} \ ; \ \mathsf{Writes}_x \subseteq \mathsf{VIS}$ for any $x \in X$, and $\rho(\mathcal{T}, \mathsf{VIS}) \ ; \ \mathsf{AR} \ ; \ \pi(\mathcal{T}, \mathsf{VIS}) \subseteq \mathsf{VIS}$. Since $\Sigma = \{(\rho_{\mathsf{Writes}_x}, \rho_{\mathsf{Writes}_x})\}_{x \in X} \cup \{(\rho, \pi)\}$, we have proved that $\mathcal{P} \in \mathsf{PreExecutions}(\Sigma)$. Let now $\mathcal{G}' = \mathsf{graph}(\mathcal{P})$. The proof that $\mathcal{G}'$ is a well-defined dependency graph is analogous to the one given for abstract executions in [14, extended version, Proposition 23].

It remains to prove that $\mathcal{G}' = \mathcal{G}$; to this end, it suffices to show that for any $x \in \mathsf{Obj}$, $\mathsf{WR}_{\mathcal{G}}(x) = \mathsf{WR}_{\mathcal{G}'}(x)$, and $\mathsf{WW}_{\mathcal{G}}(x) = \mathsf{WW}_{\mathcal{G}'}(x)$.

Let T, S be two entities such that $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$. By definition, $S \ni \mathtt{read} \ x : n$, and $T \ni \mathtt{write} \ x : n$ for some n. Also, let $T' \ni \mathtt{write} \ x : n$ be the entity such that $T' \xrightarrow{\mathsf{WR}_{\mathcal{G}'}(x)} S$, which exists because $S \ni \mathtt{read} \ x : n$ and $\mathcal{G}'$ is a well-defined dependency graph. By definition, $T' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(S) \cap \mathsf{Writes}_x)$, and in particular $T' \xrightarrow{\mathsf{VIS}} S$.

Since $T, T' \ni \mathtt{write} \ x : n$, we have that either $T = T'$, $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T'$, or $T' \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T$:

- if $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T'$, then by definition, the edges $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$ and $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T'$ induce the anti-dependency $S \xrightarrow{\mathsf{RW}_{\mathcal{G}}(x)} T'$. However, now we have that $T' \ni \mathtt{write} \ x : \_$, $T' \xrightarrow{\mathsf{VIS}} S$ and $S \xrightarrow{\mathsf{RW}_{\mathcal{G}}(x)} T'$: by the inequation (A3), it follows that $T' \xrightarrow{\mathsf{AR}} T'$, contradicting the assumption that $\mathsf{AR} \cap \mathsf{Id} \subseteq \emptyset$,
- if $T' \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T$, then note that by the inequation (A1) it has to be $T' \xrightarrow{\mathsf{AR}} T$; also, because of the dependency $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$ and the inequality (V1), it has to be $T \xrightarrow{\mathsf{VIS}} S$; but this contradicts the assumption that $T' = \max_{\mathsf{AR}}(\mathsf{VIS}^{-1}(S) \cap \mathsf{Writes}_x)$.

We are left with the case $T = T'$, from which $T \xrightarrow{\mathsf{WR}_{\mathcal{G}'}(x)} S$ follows.

Next, suppose that $T' \xrightarrow{\mathsf{WR}_{\mathcal{G}'}(x)} S$. Then $S \ni \mathtt{read} \ x : n$ for some n, and because $\mathcal{G}$ is a dependency graph, there exists an entity T such that $T \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} S$. We can proceed as in the previous case to show that $T = T'$, hence $T' \xrightarrow{\mathsf{WR}_{\mathcal{G}}(x)} T$.

Finally, we need to show that $\mathsf{WW}_{\mathcal{G}'}(x) = \mathsf{WW}_{\mathcal{G}}(x)$. First, note that if $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} S$, then $T, S \in \mathsf{Writes}_x$. By the inequation (A1) we obtain that $T \xrightarrow{\mathsf{AR}} S$, so that $T \xrightarrow{\mathsf{WW}_{\mathcal{G}'}(x)} S$ by definition of $\mathsf{graph}(\mathcal{P})$.

If $T \xrightarrow{\mathsf{WW}_{\mathcal{G}'}(x)} S$, then it has to be the case that $T \xrightarrow{\mathsf{AR}} S$, $T, S \in \mathsf{Writes}_x$. Since $\mathsf{WW}_{\mathcal{G}}(x)$ is total over Writes_x, then either $T = S$, $S \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} T$ or $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} S$. However, the first case is not possible because it would imply $T \xrightarrow{\mathsf{AR}} T$, contradicting the assumption that $\mathsf{AR} \cap \mathsf{Id} \subseteq \varnothing$. The second case is not possible either, because by the inequality (A1) we would get that $S \xrightarrow{\mathsf{AR}} T \xrightarrow{\mathsf{AR}} S$, and by the inequality (A4) $S \xrightarrow{\mathsf{AR}} S$, again contradicting the assumption that $\mathsf{AR} \cap \mathsf{Id} \subseteq \varnothing$. We are left with $T \xrightarrow{\mathsf{WW}_{\mathcal{G}}(x)} S$, as we wanted to prove.

The fact that $\mathsf{RW}_{\mathcal{G}} = \mathsf{RW}_{\mathcal{G}'}$ follows from the observation that, for any object $x \in \mathsf{Obj}$, $\mathsf{RW}_{\mathcal{G}}(x) = \mathsf{WR}_{\mathcal{G}}^{-1}(x) ; \mathsf{WW}_{\mathcal{G}}(x) = \mathsf{WR}_{\mathcal{G}'}^{-1}(x) ; \mathsf{WW}_{\mathcal{G}'}(x) = \mathsf{RW}_{\mathcal{G}'}(x)$. ◂

D.2 Proof of Proposition 21

In the following, we let $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$, and we assume that $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_{\Sigma(\mathcal{G})}$ such that $\mathsf{AR} \cap \mathsf{Id} = \varnothing$. Also, we assume that there exist two transactions T, S such that $T \neq S$, $\neg(T \xrightarrow{\mathsf{AR}} S)$, and $\neg(S \xrightarrow{\mathsf{AR}} T)$. The proof of Proposition 21 is a direct consequence of the following result, which we will prove in this section:

▸ Proposition 48. Define the following relations:

- $\partial A = \{(T, S)\}$,
- $\Delta A = \mathsf{AR}? ; \partial A ; \mathsf{AR}?$,
- $\mathsf{AR}_\nu = \mathsf{AR} \cup \Delta\mathsf{AR}$,
- $\partial V = \rho(\mathsf{VIS}) ; \Delta A ; \pi(\mathsf{VIS})$,
- $\Delta V = \mathsf{VIS}? ; \partial V ; \mathsf{VIS}?$,
- $\mathsf{VIS}_\nu = \mathsf{VIS} \cup \Delta V$,
- $\mathsf{AntiVIS}_\nu = \mathsf{VIS}_\nu? ; \mathsf{RW} ; \mathsf{VIS}_\nu?$.

Then $(X_V = \mathsf{VIS}_\nu, X_A = \mathsf{AR}_\nu, X_N = \mathsf{AntiVIS}_\nu)$ is a solution to $\mathsf{System}_\Sigma(\mathcal{G})$. Furthermore, it is the smallest solution for which the relation corresponding to the unknown X_A contains the relation $(\mathsf{AR} \cup \partial A)$.

Before proving Proposition 48, we need to prove several technical lemmas.

▸ **Lemma 49** (∂-Cut). *For any relations $R, P, Q \subseteq \mathcal{T} \times \mathcal{T}$ we have that $(R ; \partial A ; Q ; \partial A ; P) \subseteq (R ; \partial A ; P)$, and $(R ; \partial V ; Q ; \partial V ; P) \subseteq (R ; \partial V ; P)$.*

Proof. Recall that $\partial A = \{(T, S)\}$, where T, S are not related by AR. That is, whenever $T'' \xrightarrow{\partial A} S''$, for some $T'', S'' \in \mathcal{T}$, then $T'' = T, S'' = S$. It follows that $(T', S') \in (R ; \partial A ; Q ; \partial A ; P)$ if and only if $T' \xrightarrow{R} T \xrightarrow{\partial A} S \xrightarrow{Q} T \xrightarrow{\partial A} S \xrightarrow{P} S'$. As a consequence, $T' \xrightarrow{R} T \xrightarrow{\partial A} S \xrightarrow{P} S'$, as we wanted to prove.

Next, recall that $\partial V = \rho(\mathsf{VIS}) ; \Delta A ; \pi(\mathsf{VIS})$, where $\Delta A = \mathsf{AR}? ; \partial A ; \mathsf{AR}?$. That is, $\partial V = \rho(\mathsf{VIS}) ; \mathsf{AR}? ; \partial A ; \mathsf{AR}? ; \pi(\mathsf{VIS})$. If we apply the statement above to the relations $R' := (R ; \rho(\mathsf{VIS}) ; \mathsf{AR}?)$, $Q' := (\mathsf{AR}? ; \pi(\mathsf{VIS}) ; Q ; \rho(\mathsf{VIS}) ; \mathsf{AR}?)$, $P' := (\mathsf{AR}? ; \pi(\mathsf{VIS}) ; P)$, we obtain that

$$\begin{array}{rcl} R ; \partial V ; Q ; \partial V ; P & = & (R ; \rho(\mathsf{VIS}) ; \mathsf{AR}?) ; \partial A ; (\mathsf{AR}? ; \pi(\mathsf{VIS}) ; Q ; \\ & & \rho(\mathsf{VIS}) ; \mathsf{AR}?) ; \partial A ; (\mathsf{AR}? ; \pi(\mathsf{VIS}) ; P) \\ & = & R' ; \partial A ; Q' ; \partial A ; P' \\ & \subseteq & R' ; \partial A ; P' \\ & = & R ; \rho(\mathsf{VIS}) ; \mathsf{AR}? ; \partial A ; \mathsf{AR}? ; \pi(\mathsf{VIS}) ; P \\ & = & R ; \partial V ; P \end{array}$$

◂

▸ **Corollary 50.** *The relations AR_ν and VIS_ν are transitive.*

Proof. We only show the result for AR_ν. The statement relative to VIS_ν can be proved analogously.

It suffices to show that $\mathsf{AR}_\nu ; \mathsf{AR}_\nu = (\mathsf{AR} \cup \Delta A) ; (\mathsf{AR} \cup \Delta A) \subseteq (\mathsf{AR} \cup \Delta\mathsf{AR})$. By distributivity of ; with respect to $\cup$, this reduces to prove the following four inclusions:

- $(\mathsf{AR} ; \mathsf{AR}) \subseteq (\mathsf{AR} \cup \Delta\mathsf{AR})$. Recall that $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_\Sigma(\mathcal{G})$, hence by the inequation (A4) $\mathsf{AR} ; \mathsf{AR} \subseteq \mathsf{AR}$. It follows immediately that $\mathsf{AR} ; \mathsf{AR} \subseteq \mathsf{AR} \cup \Delta\mathsf{AR}$.
- $(\mathsf{AR} ; \Delta A) \subseteq (\mathsf{AR} \cup \Delta A)$: recall that $\Delta A = \mathsf{AR}? ; \partial A ; \mathsf{AR}?$. Because of the inequation (A4), we have that $\mathsf{AR} ; \mathsf{AR}? \subseteq \mathsf{AR}?$, Therefore

$$\begin{aligned}&\mathsf{AR} ; \Delta A = \\ &\mathsf{AR} ; (\mathsf{AR}? ; \partial A ; \mathsf{AR}?) = \\ &\mathsf{AR}? ; \partial A ; \mathsf{AR}? = \\ &\Delta A \subseteq \mathsf{AR} \cup \Delta A\end{aligned}$$

- $\Delta A ; \mathsf{AR} \subseteq (\mathsf{AR} \cup \Delta A)$: This case is symmetric to the previous one.
- $(\Delta A ; \Delta A) \subseteq (\mathsf{AR} \cup \Delta A)$:

$$\begin{aligned}&\Delta A ; \Delta A = \\ &(\mathsf{AR}? ; \partial A ; \mathsf{AR}?) ; (\mathsf{AR}? ; \partial A ; \mathsf{AR}?) = \\ &\mathsf{AR}? ; \partial A ; (\mathsf{AR}? ; \mathsf{AR}?) ; \partial A ; \mathsf{AR}? \overset{\mathsf{Lem.}(49)}{\subseteq} \\ &\mathsf{AR}? ; \partial A ; \mathsf{AR}? = \\ &\Delta A \subseteq \mathsf{AR} \cup \Delta\mathsf{AR}\end{aligned}$$

where the inequation above has been obtained by applying a ∂-cut (Lemma 49).

◀

▸ **Lemma 51** (Δ-extraction (ρ case)).

$$\begin{array}{rcl}\rho(\mathsf{VIS}_\nu) & \subseteq & \rho(\mathsf{VIS}) \cup (\mathsf{VIS}? ; \rho(\mathsf{VIS}) ; \Delta A) \\ \rho(\mathsf{VIS}_\nu) & \subseteq & \rho(\mathsf{VIS}) \cup (\Delta A ; \pi(\mathsf{VIS}) ; \mathsf{VIS}?) .\end{array}$$

We refer to the first inequality as right Δ-extraction*, and to the second inequality as* left Δ-extraction.

▸ **Lemma 52** (Δ-extraction (π case)).

$$\begin{array}{rcl}\pi(\mathsf{VIS}_\nu) & \subseteq & \pi(\mathsf{VIS}) \cup (\mathsf{VIS}? ; \rho(\mathsf{VIS}) ; \Delta A) \\ \pi(\mathsf{VIS}_\nu) & \subseteq & \pi(\mathsf{VIS}) \cup (\Delta A ; \pi(\mathsf{VIS}) ; \mathsf{VIS}?) .\end{array}$$

Proof. We only show how to prove the first inequation of Lemma 51. The proof of the second inequation of Lemma 51, and the proof of Lemma 52, are similar.

Recall that $\mathsf{VIS}_\nu = \mathsf{VIS} \cup \Delta V$. By Proposition 26(iii), we have that

$$\rho(\mathsf{VIS}_\nu) = \rho(\mathsf{VIS}) \cup \rho(\Delta V),$$

by unfolding the definition of specification function to the RHS, and by applying the distributivity of $\cap$ over $\cup$, we get

$$\rho(\mathsf{VIS}_\nu) = (\rho(\mathcal{T} \times \mathcal{T}) \cap \mathsf{VIS}?) \cup (\rho(\mathcal{T} \times \mathcal{T}) \cap \Delta V?) = \rho(\mathcal{T} \times \mathcal{T}) \cap (\mathsf{VIS}? \cup \Delta V?)$$

Note that for any relation R_1, R_2, $R_1? \cup R_2? = R_1? \cup R_2$, hence we can elide the reflexive closure in the term $(\Delta V)?$ of the equality above

$$\rho(\mathsf{VIS}_\nu) = \rho(\mathcal{T} \times \mathcal{T}) \cap (\mathsf{VIS}? \cup \Delta V)$$

By applying the distributivity of $\cap$ over $\cup$, and then by applying the definition of specification function, we get

$$\rho(\mathsf{VIS}_\nu) = (\rho(\mathcal{T} \times \mathcal{T}) \cap \mathsf{VIS}?) \cup (\rho(\mathcal{T} \times \mathcal{T}) \cap \Delta V) = \\ \rho(\mathsf{VIS}) \cup (\rho(\mathcal{T} \times \mathcal{T}) \cap \Delta V) \subseteq \rho(\mathsf{VIS}) \cup (\Delta V)$$

Because $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_\Sigma(\mathcal{G})$, by Equation (A2) we obtain that $\mathsf{VIS}? \subseteq \mathsf{AR}?$. Also, by Proposition 26(i) we have that $\pi(\mathsf{VIS}) \subseteq \mathsf{VIS}? \subseteq \mathsf{AR}?$. Finally, the inequation (A4)states that $\mathsf{AR} \,;\, \mathsf{AR} \subseteq \mathsf{AR}$, from which $\mathsf{AR}? \,;\, AR? \subseteq \mathsf{AR}?$ follows. By putting all these together, we get

$$\rho(\mathsf{VIS}_\nu) \subseteq \rho(\mathsf{VIS}) \cup \Delta V = \\ \rho(\mathsf{VIS}) \cup (\mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, \Delta A \,;\, \pi(\mathsf{VIS}) \,;\, \mathsf{VIS}?) \subseteq \\ \rho(\mathsf{VIS}) \cup (\mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, (\mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}?) \,;\, \mathsf{AR}? \,;\, \mathsf{AR}?) \\ \rho(\mathsf{VIS}) \cup (\mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, (\mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}?)) = \rho(\mathsf{VIS}) \cup (\mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, \Delta A).$$

as we wanted to prove. ◂

▸ **Lemma 53.**

$$\begin{array}{lcl} \partial\mathsf{VIS} & \subseteq & \Delta A \,;\, \pi(\mathsf{VIS}) \\ \partial\mathsf{VIS} & \subseteq & \rho(\mathsf{VIS}) \,;\, \Delta A \end{array}$$

Proof. Recall that $\partial V = \rho(\mathsf{VIS}) \,;\, \Delta A \,;\, \pi(\mathsf{VIS})$. We prove the first inequality as follows:

$$\begin{array}{lcl} \Delta V & = & \rho(\mathsf{VIS}) \,;\, \Delta A \,;\, \pi(\mathsf{VIS}) \\ & = & \rho(\mathsf{VIS}) \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \pi(\mathsf{VIS}) \\ & \subseteq & \mathsf{AR}? \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \pi(\mathsf{VIS}) \\ & \subseteq & \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \pi(\mathsf{VIS}) \\ & = & \Delta A \,;\, \pi(\mathsf{VIS}) \end{array}$$

where we have used the fact that $\rho(\mathsf{VIS}) = \rho(\mathcal{T} \times \mathcal{T}) \cap \mathsf{VIS}? \subseteq \mathsf{VIS}? \subseteq \mathsf{AR}?$, because of the definition of specification function and because of Inequation (A2). ◂

The next step needed to prove Proposition 48 is that of verifying that by substituting AR_ν for X_A, VIS_ν for X_V, and $\mathsf{AntiVIS}_\nu$ for X_N, each of the inequations in $\mathsf{System}_\Sigma(\mathcal{G})$ is satisfied. The next propositions show that this is indeed the case.

▸ Proposition 54.

$$\mathsf{VIS}_\nu \subseteq \mathsf{AR}_\nu$$

Proof. Recall that $\mathsf{VIS}_\nu = \mathsf{VIS} \cup \Delta V$, $\mathsf{AR}_\nu = \mathsf{AR} \cup \Delta A$. To prove that $\mathsf{VIS}_\nu \subseteq \mathsf{AR}_\nu$, it suffices to show that $\mathsf{VIS} \subseteq (\mathsf{AR} \cup \Delta A)$, and $\Delta V \subseteq (\mathsf{AR} \cup \Delta A)$.

The inequation $\mathsf{VIS} \subseteq \mathsf{AR} \cup \Delta A$ follows immediately the fact that $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_\Sigma(\mathcal{G})$, and from the inequation (A2)- $\mathsf{VIS} \subseteq \mathsf{AR}$.

It remains to prove that $\Delta V \subseteq \mathsf{AR} \cup \Delta A$. In fact, we prove a stronger result, namely $\Delta V \subseteq \Delta A$. This is done as follows:

$$\Delta V = \mathsf{VIS}? \,;\, \partial V \,;\, \mathsf{VIS}? = \mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, \Delta A \,;\, \pi(\mathsf{VIS}) \,;\, \mathsf{VIS}? = \\ \mathsf{VIS}? \,;\, \rho(\mathsf{VIS}) \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \pi(\mathsf{VIS}) \,;\, \mathsf{VIS}? \subseteq \\ \mathsf{VIS}? \,;\, \mathsf{VIS}? \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \mathsf{VIS}? \,;\, \mathsf{VIS}? \overset{\text{(V2)}}{\subseteq} \\ \mathsf{VIS}? \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \mathsf{VIS}? \overset{\text{(A2)}}{\subseteq} \\ \mathsf{AR}? \,;\, \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? \,;\, \mathsf{AR}? \overset{\text{(A4)}}{\subseteq} \\ \mathsf{AR}? \,;\, \partial A \,;\, \mathsf{AR}? = \Delta A,$$

where we used the notation $R_1 \overset{(eq)}{\subseteq} R_2$ to denote that the inequation $R_1 \subseteq R_2$ follows from the Inequation (eq), taken from Figure 4. ◂

▸ Proposition 55.

$$\rho(\mathsf{VIS}_\nu) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}_\nu) \subseteq \mathsf{VIS}_\nu.$$

Proof. First, we perform a right Δ-extraction (Lemma 51) of $\rho(\mathsf{VIS}_\nu)$, and a left Δ-extraction (Lemma 52) of $\pi(\mathsf{VIS}_\nu)$. This gives us the following inequation:

$$\rho(\mathsf{VIS}_\nu) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}_\nu) \subseteq (\rho(\mathsf{VIS}) \cup (\mathsf{VIS?} \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} (\pi(\mathsf{VIS}) \cup (\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?})$$

and we rewrite the RHS of the above by applying the distributivity of $\cup$ over ; .

$$\begin{aligned} \rho(\mathsf{VIS}_\nu) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}_\nu) \subseteq \quad & \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) \cup \\ & \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} (\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?}) \cup \\ & \mathsf{VIS?} \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) \cup \\ & \mathsf{VIS?} \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?}. \end{aligned}$$

We show that each of the components of the union of the RHS of the inequation above is included in VIS_ν, from which we get the desired result $\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) \subseteq \mathsf{VIS}_\nu$.

- $\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) \subseteq \mathsf{VIS}_\nu$. Recall that $\mathsf{AR}_\nu = \mathsf{AR} \cup \Delta A$, from which we get that

  $$\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) = (\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS})) \cup \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}).$$

  We prove that each of the components of the union in the RHS above are included in VIS_ν. First, observe that

  $$\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) \subseteq \mathsf{VIS} \subseteq (\mathsf{VIS} \cup \Delta V) = \mathsf{VIS}_\nu$$

  because of Inequation (V4). Also, we have that

  $$\rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) = \partial V \subseteq \mathsf{VIS?} \mathbin{;} \partial V \mathbin{;} \mathsf{VIS?} = \Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu$$

  and in this case there is nothing left to prove.
- $\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} (\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?}) \subseteq \mathsf{VIS}_\nu$. Again, by unfolding the definition of AR_ν and by applying the distributivity of $\cup$ over ; , we obtain that

  $$\begin{aligned} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?} = \quad & \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?} \cup \\ & \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS?} \end{aligned}$$

We prove that each of the components of the union in the RHS above is included in VIS_ν.

$$\begin{aligned}
&\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR} \mathbin{;} (\mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}?) \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \overset{\text{(A4)}}{\subseteq} \\
&\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\partial V \mathbin{;} \mathsf{VIS}? \subseteq \mathsf{VIS}? \mathbin{;} \partial V \mathbin{;} \mathsf{VIS}? = \\
&\Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu
\end{aligned}$$

$$\begin{aligned}
&\rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \overset{\mathsf{Lem}.49}{\subseteq} \\
&\rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\partial V \mathbin{;} \mathsf{VIS}? \subseteq \mathsf{VIS}? \mathbin{;} \partial V \mathbin{;} \mathsf{VIS}? = \Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu.
\end{aligned}$$

- $\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) \subseteq \mathsf{VIS}_\nu$. As for the two cases above, we unfold AR_ν and distribute the resulting union over ; : this leads to

$$\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \pi(\mathsf{VIS}) = \qquad \begin{aligned}&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) \cup \\ &\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}).\end{aligned}$$

Then we prove that each of the two terms in the union on the RHS above is included in VIS_ν:

$$\begin{aligned}
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \mathsf{AR} \mathbin{;} \pi(\mathsf{VIS}) \overset{\text{(A4)}}{\subseteq} \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \partial V \subseteq \\
&\mathsf{VIS}? \mathbin{;} \partial V \mathbin{;} \mathsf{VIS}? = \Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu
\end{aligned}$$

$$\begin{aligned}
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} AR? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \overset{\mathsf{Lem}.49}{\subseteq} \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \rho(\mathsf{VIS}) = \\
&\mathsf{VIS}? \mathbin{;} \partial V \subseteq \\
&\mathsf{VIS}? \mathbin{;} \partial V \mathbin{;} \mathsf{VIS}? = \Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu
\end{aligned}$$

- $\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}?$ in this case we have the following:

$$\begin{aligned}
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \mathsf{AR}_\nu \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \overset{\mathsf{Lem}.49}{\subseteq} \\
&\mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? = \\
&\mathsf{VIS}? \mathbin{;} \partial V \mathbin{;} \mathsf{VIS}? = \Delta V \subseteq \mathsf{VIS} \cup \Delta V = \mathsf{VIS}_\nu
\end{aligned}$$

◀

▸ Proposition 56.

$$\Big(\pi(\mathsf{VIS}_\nu) \ ; \mathsf{AntiVIS}_\nu \ ; \rho(\mathsf{VIS}_\nu)\Big)\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu.$$

Proof. Recall that $\mathsf{AntiVIS}_\nu = \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_{\nu?}$. Thus, we need to prove that

$$\Big(\pi(\mathsf{VIS}_\nu) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_{\nu?} \ ; \rho(\mathsf{VIS}_\nu)\Big)\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu.$$

We start by performing a Δ-extraction both for the specification functions π and ρ:

$$\begin{aligned}\Big(\pi(\mathsf{VIS}_\nu) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_{\nu?} \ ; \rho(\mathsf{VIS}_\nu)\Big)\backslash\mathsf{Id} \subseteq \\ (\pi(\mathsf{VIS}) \cup (\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}?)) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; ((\mathsf{VIS}? \ ; \rho(\mathsf{VIS}) \ ; \Delta A) \cup \rho(\mathsf{VIS}))) \ \backslash\mathsf{Id} = \\ \\ (\pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id}\cup \\ (\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}? \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id}\cup \\ (\pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \mathsf{VIS}? \ ; \rho(\mathsf{VIS}) \ ; \Delta A)\backslash\mathsf{Id}\cup \\ (\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}? \ ; \mathsf{VIS}_{\nu?} \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \mathsf{VIS}? \ ; \rho(\mathsf{VIS}) \ ; \Delta A)\backslash\mathsf{Id}\end{aligned}$$

We prove that each of the four terms of the union above is included in AR_ν. To this end, it suffices to prove the following:

$$(\pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu? \tag{60}$$

In fact, if the inequation (60) is satisfied, we obtain that

- $(\pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu$:

$$\begin{aligned}&(\pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \overset{(60)}{\subseteq} \\ &\mathsf{AR}_\nu?\backslash\mathsf{Id} = (\mathsf{AR}_\nu \cup \mathsf{Id})\backslash\mathsf{Id} = \mathsf{AR}_\nu\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu,\end{aligned}$$

- $(\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}? \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu$:

$$\begin{aligned}&(\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}? \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \subseteq \\ &(\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \overset{\mathsf{Cor.}(50)}{\subseteq} \\ &(\Delta A \ ; \pi(\mathsf{VIS}) \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \ ; \rho(\mathsf{VIS}))\backslash\mathsf{Id} \overset{(60)}{\subseteq} \\ &(\Delta A \ ; \mathsf{AR}_\nu?)\backslash\mathsf{Id} = \\ &(\Delta A \ ; (\mathsf{AR} \cup \Delta A)?)\backslash\mathsf{Id} = \\ &(\Delta A \ ; (\mathsf{AR}? \cup \Delta A))\backslash\mathsf{Id} = \\ &(\Delta A \ ; \mathsf{AR}?)\backslash\mathsf{Id} \cup (\Delta A \ ; \Delta A)\backslash\mathsf{Id} = \\ &(\mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}? \ ; \mathsf{AR}?)\backslash\mathsf{Id} \cup (\mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}? \ ; \mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}?)\backslash\mathsf{Id} \overset{\mathsf{Lem}.49}{\subseteq} \\ &(\mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}? \ ; \mathsf{AR}?)\backslash\mathsf{Id} \cup (\mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}?)\backslash\mathsf{Id} \overset{(A4)}{\subseteq} \\ &(\mathsf{AR}? \ ; \partial A \ ; \mathsf{AR}?)\backslash\mathsf{Id} = (\Delta A)\backslash\mathsf{Id} \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu\end{aligned}$$

- $(\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu$:

$$\begin{aligned}&(\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \subseteq\\ &(\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \overset{\mathsf{Cor.}(50)}{\subseteq}\\ &(\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \overset{(60)}{\subseteq} (\mathsf{AR}_\nu? \mathbin{;} \Delta A)\backslash\mathsf{Id} =\\ &((\mathsf{AR} \cup \Delta A)? \mathbin{;} \Delta A)\backslash\mathsf{Id} =\\ &((\mathsf{AR}? \cup \Delta A) \mathbin{;} \Delta A)\backslash\mathsf{Id} =\\ &(\mathsf{AR}? \mathbin{;} \Delta A)\backslash\mathsf{Id} \cup (\Delta A \mathbin{;} \Delta A)\backslash\mathsf{Id} \subseteq\\ &(\Delta A)\backslash\mathsf{Id} \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu\end{aligned}$$

- $(\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \subseteq \mathsf{AR}_\nu$: here it suffices to apply a ∂-cut (Lemma 49) to obtain the result:

$$\begin{aligned}&(\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A)\backslash\mathsf{Id} \subseteq\\ &\Delta A \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \Delta A =\\ &\mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \mathbin{;} \pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? \overset{\mathsf{Lem.}(49)}{\subseteq}\\ &\mathsf{AR}? \mathbin{;} \partial A \mathbin{;} \mathsf{AR}? = \Delta A \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu\end{aligned}$$

Let then prove Inequation (60): we have that

$$\begin{aligned}&\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}_\nu? \mathbin{;} \rho(\mathsf{VIS}) =\\ &\pi(\mathsf{VIS}) \mathbin{;} (\mathsf{VIS} \cup \Delta V)? \mathbin{;} \mathsf{RW} \mathbin{;} (\mathsf{VIS} \cup \Delta V)? \mathbin{;} \rho(\mathsf{VIS}) =\\ \\ &\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \cup\\ &\pi(\mathsf{VIS}) \mathbin{;} \Delta V \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \cup\\ &\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{RW} \mathbin{;} \Delta V \mathbin{;} \rho(\mathsf{VIS}) \cup\\ &\pi(\mathsf{VIS}) \mathbin{;} \Delta V \mathbin{;} \mathsf{RW} \mathbin{;} \rho(\mathsf{VIS}) \mathbin{;} \rho(\mathsf{VIS})\end{aligned}$$

We prove that each of the terms in the union above is included in $\mathsf{AR}_\nu?$.

-

$$\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \subseteq \mathsf{AR}_\nu? \tag{61}$$

$$\begin{aligned}&\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}) \subseteq\\ &((\pi(\mathsf{VIS}) \mathbin{;} \mathsf{VIS}? \mathbin{;} \mathsf{RW} \mathbin{;} \mathsf{VIS}? \mathbin{;} \rho(\mathsf{VIS}))\backslash\mathsf{Id}) \cup \mathsf{Id} \overset{(N1),(N2),(N3)}{\subseteq}\\ &((\pi(\mathsf{VIS}) \mathbin{;} \mathsf{AntiVIS} \mathbin{;} \rho(\mathsf{VIS}))\backslash\mathsf{Id}) \cup \mathsf{Id} \overset{(A5)}{\subseteq}\\ &\mathsf{AR} \cup \mathsf{Id} \subseteq \mathsf{AR} \cup \Delta A \cup \mathsf{Id} = \mathsf{AR}_\nu \cup \mathsf{Id} = \mathsf{AR}_\nu?\end{aligned}$$

- $\pi(\mathsf{VIS})\ ;\ \Delta V\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) \subseteq \mathsf{AR}_\nu?$:

$$
\begin{aligned}
&\pi(\mathsf{VIS})\ ;\ \Delta V\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) = \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) = \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS})\ ;\ \Delta A\ ;\ \pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) \subseteq \\
&\mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \Delta A\ ;\ \pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) \overset{\text{(A2),(A4)}}{\subseteq} \\
&\mathsf{AR}?\ ;\ \Delta A\ ;\ \pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) \overset{(61)}{\subseteq} \\
&\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}_\nu? = \\
&\mathsf{AR}?\ ;\ \Delta A\ ;\ (\mathsf{AR}? \cup \Delta A) = \\
&(\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}?) \cup (\mathsf{AR}?\ ;\ \Delta A\ ;\ \Delta A) = \\
&(\mathsf{AR}?\ ;\ \mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?) \cup (\mathsf{AR}?\ ;\ \mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?\ ;\ \mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?) \overset{\text{(A4)}}{\subseteq} \\
&(\mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?) \cup (\mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?) \overset{\mathsf{Lem.}(49)}{\subseteq} \\
&(\mathsf{AR}?\ ;\ \partial A\ ;\ \mathsf{AR}?) = \Delta A \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu \subseteq \mathsf{AR}_\nu?
\end{aligned}
$$

- $\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \Delta V\ ;\ \rho(\mathsf{VIS}) \subseteq \mathsf{AR}_\nu?$:

$$
\begin{aligned}
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \Delta V\ ;\ \rho(\mathsf{VIS}) = \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) = \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS})\ ;\ \Delta A\ ;\ \rho(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS}) \subseteq \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS})\ ;\ \Delta A\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}? \overset{\text{(A2),(A4)}}{\subseteq} \\
&\pi(\mathsf{VIS})\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \rho(\mathsf{VIS})\ ;\ \Delta A\ ;\ \mathsf{AR}? \overset{(61)}{\subseteq} \\
&\mathsf{AR}_\nu?\ ;\ \Delta A\ ;\ \mathsf{AR}? = \\
&(\mathsf{AR}?\ ;\ \Delta A)\ ;\ \Delta A\ ;\ \mathsf{AR}? = \\
&(\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}?) \cup (\Delta A\ ;\ \Delta A\ ;\ \mathsf{AR}?) \overset{\mathsf{Lem.}(49)}{\subseteq} \\
&(\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}?) \cup (\Delta A\ ;\ \mathsf{AR}?) = \\
&(\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}?) = \Delta A \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu \subseteq \mathsf{AR}_\nu?
\end{aligned}
$$

- $\pi(\mathsf{VIS})\ ;\ \Delta V\ ;\ \mathsf{RW}\ ;\ \Delta V\ ;\ \rho(\mathsf{VIS}) \subseteq \mathsf{AR}_\nu?$

$$
\begin{aligned}
&\pi(\mathsf{VIS})\ ;\ \Delta V\ ;\ \mathsf{RW}\ ;\ \Delta V\ ;\ \rho(\mathsf{VIS}) \subseteq \\
&\mathsf{VIS}?\ ;\ \Delta V\ ;\ \mathsf{RW}\ ;\ \Delta V\ ;\ \mathsf{VIS}? = \\
&\mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}?\ ;\ \mathsf{RW}\ ;\ \mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}? \overset{\mathsf{Lem.}(49)}{\subseteq} \\
&\mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}? \overset{\text{(V2)}}{\subseteq} \\
&\mathsf{VIS}?\ ;\ \partial V\ ;\ \mathsf{VIS}? \overset{\text{(A2)}}{\subseteq} \\
&\mathsf{VIS}?\ ;\ \mathsf{VIS}?\ ;\ \Delta A\ ;\ \mathsf{VIS}?\ ;\ \mathsf{VIS}? \overset{\text{(V2),(A2)}}{\subseteq} \\
&\mathsf{AR}?\ ;\ \Delta A\ ;\ \mathsf{AR}? = \Delta A \subseteq \mathsf{AR} \cup \Delta A = \mathsf{AR}_\nu \subseteq \mathsf{AR}_\nu?
\end{aligned}
$$

◂

▸ Proposition 57.

$$
\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x]\ ;\ \mathsf{VIS}_\nu\ ;\ \mathsf{RW}(x) \subseteq \mathsf{AR}_\nu.
$$

Proof. Let T', U, S' be such that $T' \in \mathsf{Writes}_x$, $T' \xrightarrow{\mathsf{VIS}_\nu} U \xrightarrow{\mathsf{RW}(x)} S'$ for some object $x \in \mathsf{Obj}$. We need to show that $T' \xrightarrow{\mathsf{AR}_\nu} S'$. By definition, $\mathsf{VIS}_\nu = \mathsf{VIS} \cup \Delta V$. Thus, $T' \xrightarrow{\mathsf{VIS}} U$ or $T' \xrightarrow{\Delta V} U$. If $T' \xrightarrow{\mathsf{VIS}} U$, then $T' \xrightarrow{\mathsf{VIS}} U \xrightarrow{\mathsf{RW}(x)} S'$ and $T' \in \mathsf{Writes}_x$. By the inequation (A3) we have that $T' \xrightarrow{\mathsf{AR}} S'$, which implies the desired $T' \xrightarrow{\mathsf{AR}_\nu} S'$.

Suppose then that $T' \xrightarrow{\Delta V} U$. By unfolding the definition of ΔV, we have that

$$T' \xrightarrow{\mathsf{VIS}?;\rho(\mathsf{VIS})} T'' \xrightarrow{\mathsf{AR}?} T \xrightarrow{\partial A} S \xrightarrow{\mathsf{AR}?} U' \xrightarrow{\pi(\mathsf{VIS});\mathsf{VIS}?} U \xrightarrow{\mathsf{RW}(x)} S'.$$

Recall that by definition of ∂A, the transactions T and S are not related by AR. Note that, since $U \xrightarrow{\mathsf{RW}(x)} S'$, then $U \in \mathsf{Reads}_x, S' \in \mathsf{Writes}_x$. Recall that $\mathsf{WW}(x)$ is a total order over Writes_x. Therefore, we have three possible cases: $T' \xrightarrow{\mathsf{WW}(x)} S'$, $T' = S'$ or $T' \xrightarrow{\mathsf{WW}(x)} S'$. These cases are analysed separately.

- $T' \xrightarrow{\mathsf{WW}(x)} S'$: by the inequality (A1)we have that $T' \xrightarrow{\mathsf{AR}} S'$. Thus, $T' \xrightarrow{\mathsf{AR}_\nu} S'$.
- $T' = S'$: this case is not possible. We first prove that $U' \neq T''$. Suppose $U' = T''$. Then $S \xrightarrow{\mathsf{AR}?} U' = T'' \xrightarrow{\mathsf{AR}?} T$, that is $S \xrightarrow{\mathsf{AR}?} T$. But by hypothesis, T and S are not related by AR, hence we get a contradiction.
  Let then $U' \neq T''$. Since we have
  $$U' \xrightarrow{\pi(\mathsf{VIS});\mathsf{VIS}?} U \xrightarrow{\mathsf{RW}(x)} S' = T' \xrightarrow{\mathsf{VIS}?;\rho(\mathsf{VIS})} T''$$
  we have that $U' \xrightarrow{\mathsf{AR}} T''$ by the inequality (A5). Thus, $S \xrightarrow{\mathsf{AR}?} U' \xrightarrow{\mathsf{AR}} T'' \xrightarrow{\mathsf{AR}?} T$, or equivalently $S \xrightarrow{\mathsf{AR}} T$. Again, this contradict the assumption that S and T are not related by AR.
- $S' \xrightarrow{\mathsf{WW}(x)} T'$: this case is also not possible. Recall that $U \xrightarrow{\mathsf{RW}(x)} S'$; that is, there exists an entity U'' such that $U'' \xrightarrow{\mathsf{WR}(x)} U$, $U'' \xrightarrow{\mathsf{WW}(x)} S'$. By the transitivity of $\mathsf{WW}(x)$, we have that $U'' \xrightarrow{\mathsf{WW}(x)} T'$. Thus, $U \xrightarrow{\mathsf{RW}(x)} T'$. We can proceed as in the case above to show that this implies $S \xrightarrow{\mathsf{AR}} T$, contradicting the assumption that T and S are not related by AR.

◂

Finally, we prove the following:

▸ Proposition 58. The triple $(X_V = \mathsf{VIS}_\nu, X_A = \mathsf{AR}_\nu, X_N = \mathsf{AntiVIS}_\nu)$ is included in the least solution to $\mathsf{System}_\Sigma(\mathcal{G})$ for which the relation corresponding to the unknown X_A includes the relation $\mathsf{AR} \cup \partial A$.

Proof.. Let $(X_V = \mathsf{VIS}', X_A = \mathsf{AR}', X_N = \mathsf{AntiVIS}')$ be a solution to $\mathsf{System}_\Sigma(\mathcal{G})$ such that $(\mathsf{AR} \cup \partial A) \subseteq \mathsf{AR}'$. We need to show that $\mathsf{AR}_\nu \subseteq \mathsf{AR}'$, $\mathsf{VIS}_\nu \subseteq \mathsf{VIS}'$, and $\mathsf{AntiVis}_\nu \subseteq \mathsf{AntiVIS}'$.

- $\mathsf{AR}_\nu \subseteq \mathsf{AR}'$: note that we have
  $$\Delta A = \mathsf{AR}? \; ; \partial A \; ; \mathsf{AR}? \subseteq \mathsf{AR}' \; ; \mathsf{AR}' \; ; \mathsf{AR}' \overset{(A4)}{\subseteq} \mathsf{AR}'$$
  from which it follows that $\mathsf{AR}_\nu = \mathsf{AR} \cup \Delta\mathsf{AR} \subseteq (\mathsf{AR}' \cup \mathsf{AR}') = \mathsf{AR}'$.
- $\mathsf{VIS}_\nu \subseteq \mathsf{VIS}'$: Observe that for any solution $(X_V = \mathsf{VIS}'', X_A = \mathsf{AR}'', X_N = \mathsf{AntiVIS}'')$ of $\mathsf{System}_\Sigma(\mathcal{G})$, the relation VIS' is determined uniquely by AR'': specifically, $\mathsf{VIS}'' = \mu V.\mathcal{F}(V, \mathsf{AR}'')$, where
  $$\mathcal{F}(V, \mathsf{AR}'') = \left(\mathsf{WR} \cup \left(\bigcup_{\{x \mid (\rho_x, \rho_x) \in \Sigma\}} \mathsf{WW}(x)\right) \cup (\rho(V) \; ; \mathsf{AR}'' \; ; \pi(V))\right)^{+}$$
  the functional $\mathcal{F}$ is monotone in its second argument, which means that the inequation $\mathsf{AR}_\nu \subseteq \mathsf{AR}'$ also implies that $\mathsf{VIS}_\nu \subseteq \mathsf{VIS}'$.

- $\mathsf{AntiVIS}_\nu \subseteq \mathsf{VIS}'$. Observe that, for any solution $(X_V = \mathsf{VIS}'', X_A = \mathsf{AR}'', X_N = \mathsf{AntiVIS}'')$, the relation $\mathsf{AntiVIS}''$ is determined uniquely by VIS''. Specifically, we have that $\mathsf{AntiVIS}'' = \mathcal{F}(\mathsf{VIS}'')$, where $\mathcal{F}(\mathsf{VIS}'') = \mathsf{VIS}''? \ ; \mathsf{RW} \ ; \mathsf{VIS}''?$. The functional $\mathcal{F}$ is monotone, from which it follows that the inequation $\mathsf{VIS}_\nu \subseteq \mathsf{VIS}'$, proved above, implies that $\mathsf{AntiVis}_\nu \subseteq \mathsf{AntiVIS}'$.

◂

Proof of Proposition 48. We need to show that $(X_V = \mathsf{VIS}_\nu, X_A = \mathsf{AR}_\nu, X_N = \mathsf{AntiVIS}_\nu)$ is a solution of $\mathsf{System}_{\mathcal{G}}(\Sigma)$. By Proposition 58, it follows that it is the smallest solution for which the relation corresponding to the unknown X_A includes $\mathsf{AR} \cup \partial A$.

Obviously we have that $\mathsf{WR} \subseteq \mathsf{VIS} \subseteq \mathsf{VIS}_\nu$, and $\bigcup\{\mathsf{WW}(x) \mid (\rho_x, \rho_x) \in \Sigma\} \subseteq \mathsf{VIS} \subseteq \mathsf{VIS}_\nu$: the inequations (V1)and (V3)are satisfied. The validity of inequation (V2) follows from Corollary 50. The inequation (V4) is also satisfied, as we have proved in Proposition 55.

The inequality (A1)is satisfied because $\mathsf{WW} \subseteq \mathsf{AR} \subseteq \mathsf{AR}_\nu$, and the inequation (A2) has been proved in Proposition 54. The validity of the inequation (A4) also follows from Corollary 50. The inequation (A5) and (A3) are satisfied, as we have proved in propositions 56 and 57.

Finally, the inequation (N1) is satisfied because $\mathsf{RW} \subseteq \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? = \mathsf{AntiVIS}_\nu$; the inequation (N2) is satisfied because $\mathsf{VIS}_\nu \ ; \mathsf{AntiVIS}_\nu = \mathsf{VIS}_\nu \ ; \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? \subseteq \mathsf{VIS}_\nu? \ ; \mathsf{RW} \ ; \mathsf{VIS}_\nu? = \mathsf{AntiVIS}_\nu$ (recall that VIS_ν is transitive by Corollary 50), and similarly we can prove that the inequation (N3) is also satisfied. ◂

D.3 Proof of Theorem 11

Throughout this section we let $\mathcal{G} = (\mathcal{T}, \mathsf{WR}, \mathsf{WW}, \mathsf{RW})$.

D.3.1 Proof of Theorem 11(1)

Recall that $\Sigma_{\mathsf{SER}} = \{(\rho_S, \rho_S)\}$, where $\rho_S(R) = \mathsf{Id}$. The instantiation of inequations (V4) and (A5), in $\mathsf{System}_{\Sigma_{\mathsf{SER}}}(\mathcal{G})$ gives rise to the inequations $X_A \subseteq X_V$ and $X_N \backslash \mathsf{Id} \subseteq X_A$.

Let $\mathsf{VIS} = \mathsf{AR} = \mathsf{AntiVIS} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+$. We prove that $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution to $\mathsf{System}_{\Sigma_{\mathsf{SER}}}(\mathcal{G})$: to this end, we show that by substituting each of the unknowns for the relation $(\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+$ in $\mathsf{System}_{\Sigma_{\mathsf{SER}}}(\mathcal{G})$, then each of the inequations of such a system is satisfied. Clearly $\mathsf{WR} \subseteq \mathsf{VIS}$, hence equation (V1) is satisfied. Because there is no consistency guarantee of the form $(\rho_x, \rho_x) \in \Sigma_{\mathsf{SER}}$, the inequation (V3) is trivially satisfied. Inequation (V2) is also satisfied. $\mathsf{VIS} \ ; \mathsf{VIS} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \ ; (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{VIS}$. Inequation (V4) requires that $\mathsf{AR} \subseteq \mathsf{VIS}$: this is also satisfied, as $\mathsf{AR} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{VIS}$.

Inequation (A1) is trivially satisfied: $\mathsf{WW} \subseteq (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{AR}$. Inequation (A2) is also satisfied: $\mathsf{VIS} = (\mathsf{SO} \cup \mathsf{WR} \cup \mathsf{RW})^+ = \mathsf{AR}$, hence $\mathsf{VIS} \subseteq \mathsf{AR}$. Inequation (A5) is satisfied as well: $\mathsf{AntiVIS} \backslash \mathsf{Id} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \backslash \mathsf{Id} \subseteq (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{AR}$. Inequation (A3) is also satisfied: $\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x] \ ; \mathsf{VIS} \ ; \mathsf{RW}(x) \subseteq \mathsf{VIS} \ ; \mathsf{RW} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \ ; \mathsf{RW} \subseteq (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{AR}$.

Inequation (N1) is obviously satisfied, as $\mathsf{RW} \subseteq (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{AntiVIS}$. For inequation (N2) , note that $\mathsf{VIS} \ ; \mathsf{AntiVIS} = (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \ ; (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ \subseteq (\mathsf{WR} \cup \mathsf{WW} \cup \mathsf{RW})^+ = \mathsf{AntiVIS}$, and it can be shown that Inequation (N3) is satisfied in a similar way.

The proof that the solution $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is the smallest solution of $\mathsf{System}_{\Sigma_{\mathsf{SER}}}(\mathcal{G})$ can be obtained as in the proof of Theorem 12. ◂

D.3.2 Proof of Theorem 11(3).

Recall that $\Sigma_{\mathsf{SI}} = \{(\rho_x, \rho_x)\}_{x \in \mathsf{Obj}} \cup \{(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})\}$, where $\rho_x(R) = [\mathsf{Writes}_x]$, $\rho_{\mathsf{SI}}(R) = R \setminus \mathsf{Id}$. By instantiating inequation (V3) to Σ_{SI} we obtain $\mathsf{WW} \subseteq X_V$, while by instantiating inequations (V4) and (A5) to the consistency guarantee $(\rho_{\mathsf{Id}}, \rho_{\mathsf{SI}})$, we obtain $X_A \ ; \ (X_V \setminus \mathsf{Id}) \subseteq X_V$, and $((X_V \setminus \mathsf{Id}) \ ; \ X_N) \setminus \mathsf{Id} \subseteq X_A$.

Let $\mathsf{AR} = ((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^+$, $\mathsf{VIS} = \mathsf{AR}? \ ; \ (\mathsf{WR} \cup \mathsf{WW})$, $\mathsf{AntiVIS} = \mathsf{VIS}? \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}?$. Then $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_{\Sigma_{\mathsf{SI}}}(\mathcal{G})$. We can prove that it is the smallest such solution in the same way as in Theorem 13.

We need to show that, by substituting $\mathsf{VIS}, \mathsf{AR}, \mathsf{AntiVIS}$ for X_V, X_A, X_N respectively, in $\mathsf{System}_{\Sigma_{\mathsf{SI}}}(\mathcal{G})$, all the inequations are satisfied. Here we give the details only for the most important of them. A full proof of this statement can be found in [14].

- $\mathsf{AR} \ ; \ (\mathsf{VIS} \setminus \mathsf{Id}) \subseteq \mathsf{VIS}$:

$$\begin{aligned}&\mathsf{AR} \ ; \ (\mathsf{VIS} \setminus \mathsf{Id}) \subseteq \mathsf{AR} \ ; \ \mathsf{VIS} = \\ &((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^+ \ ; \ ((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^* \ ; \ (\mathsf{WR} \cup \mathsf{WW}) \subseteq \\ &((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^* \ ; \ (\mathsf{WR} \cup \mathsf{WW}) = \mathsf{AR}? \ ; \ (\mathsf{WR} \cup \mathsf{WW}) = \mathsf{VIS}\end{aligned}$$

- $((\mathsf{VIS} \setminus \mathsf{Id}) \ ; \ \mathsf{AntiVIS}) \setminus \mathsf{Id} \subseteq \mathsf{AR}$:

$$\begin{aligned}&((\mathsf{VIS} \setminus \mathsf{Id}) \ ; \ \mathsf{AntiVIS}) \setminus \mathsf{Id} \subseteq \\ &\mathsf{VIS} \ ; \ \mathsf{AntiVIS} = \mathsf{VIS} \ ; \ \mathsf{VIS}? \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}? = \\ &\mathsf{VIS} \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}? = \\ &(((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^* \ ; \ (\mathsf{WR} \cup \mathsf{WW})) \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}? \subseteq \\ &((\mathsf{WR} \cup \mathsf{WW}) \ ; \ \mathsf{RW}?)^+ \ ; \ \mathsf{VIS}? = \\ &\mathsf{AR} \cup \mathsf{VIS}? \subseteq \mathsf{AR}\end{aligned}$$

  where we have used the fact that $\mathsf{AR} \ ; \ \mathsf{VIS} \subseteq \mathsf{VIS}$, which we have proved previously.

◀

D.3.3 Proof of Theorem 11(2).

Errata: the statement of the Theorem is correct. However, the proof sketch given in §5 is not. In such a proof sketch we claim that the smallest solution of the system of inequation $\mathsf{System}_{\Sigma(\mathsf{PSI})}$ is given by $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$, where $\mathsf{VIS} = (\mathsf{WR} \cup \mathsf{WW})^+$, $\mathsf{AR} = \mathsf{VIS} \cup \bigcup_{x \in \mathsf{Obj}} (\mathsf{VIS}? \ ; \ \mathsf{RW}(x))^+$, $\mathsf{AntiVIS} = \mathsf{VIS}? \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}?$. This is not true.

The correct claim for proving Theorem 11(2) is the following:

▸ Proposition 59. Let $\mathsf{VIS} = (\mathsf{WR} \cup \mathsf{WW})^+$, $\mathsf{AR} = \mathsf{VIS} \cup \bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] \ ; \ \mathsf{VIS}? \ ; \ \mathsf{RW}(x))^+$, $\mathsf{AntiVIS} = \mathsf{VIS}? \ ; \ \mathsf{RW} \ ; \ \mathsf{VIS}?$. If AR is irreflexive, then $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}(\mathcal{G})$. Furthermore, it is the smallest such solution.

In the following, we prove both Proposition 59 and Theorem 11(2).

Proof of Proposition 59. Recall that $\Sigma_{\mathsf{PSI}} = \{(\rho_x, \rho_x)\}_{x \in \mathsf{Obj}}$. Therefore, the system of inequations $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}(\mathcal{G})$ does not contain inequations (V4) and (A5), and inequation (V3) is instantiated to $\mathsf{WW} \subseteq \mathsf{VIS}$. We prove that, under the assumption that AR is irreflexive, the triple $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is a solution of $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}(\mathcal{G})$ by showing that, by substituting VIS, AR and $\mathsf{AntiVIS}$ for X_V, X_A and X_N in $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}(\mathcal{G})$, respectively, all the inequations are satisfied. The fact that the triple $(X_V = \mathsf{VIS}, X_A = \mathsf{AR}, X_N = \mathsf{AntiVIS})$ is the smallest solution of $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}(\mathcal{G})$ can be proved in the same way as in the proof of Theorem 14.

First, we observe that if AR is irreflexive, then for any $x \in \mathsf{Obj}$, $[\mathsf{Writes}_x]$; $\mathsf{VIS}?$; $\mathsf{RW}(x) \subseteq \mathsf{WW}(x)$. To see why this is true, recall that $\mathsf{WW}(x)$ is a strict, total order over Writes_x. Suppose that $T \ni \mathtt{write}\ x : \_$, $T \xrightarrow{\mathsf{VIS}?} S' \xrightarrow{\mathsf{RW}(x)} S$. Note that, since $[\mathsf{Writes}_x]$; $\mathsf{VIS}?$; $\mathsf{RW}(x) \subseteq \mathsf{AR}$, and we are assuming that the latter is irreflexive, it cannot be $T = S$. By definition of $\mathsf{RW}(x)$, $S \ni \mathtt{write}\ x : \_$. Therefore, either $T \xrightarrow{\mathsf{WW}(x)} S$, or $S \xrightarrow{\mathsf{WW}(x)} T$. However, if it were $S \xrightarrow{\mathsf{WW}(x)} T$, we would have $S \ni$ Writes_x, $S \xrightarrow{\mathsf{WW}(x)} T \xrightarrow{\mathsf{VIS}?} S' \xrightarrow{\mathsf{RW}(x)} S$: because $\mathsf{VIS} = (\mathsf{WR} \cup \mathsf{WW})^+$, $\mathsf{WW}(x)$; $\mathsf{VIS}? \subseteq \mathsf{VIS}?$, hence $S \xrightarrow{\mathsf{VIS}?} S' \xrightarrow{\mathsf{RW}(x)} S$, and because $S \ni \mathtt{write}\ x : \_$, it would follow that $S \xrightarrow{\mathsf{AR}} S$, contradicting the hypothesis that AR is irreflexive. Therefore, it must be $T \xrightarrow{\mathsf{WW}(x)} S$.

We have proved that, if AR is irreflexive, then for any $x \in \mathsf{Obj}$, $[\mathsf{Writes}_x]$; $\mathsf{VIS}?$; $\mathsf{RW}(x) \subseteq \mathsf{WW}$. An immediate consequence of this fact is the following:

$$\bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] ; \mathsf{VIS}? ; \mathsf{RW}(x))^+ \subseteq \mathsf{WW} \tag{62}$$

Next, we prove that each of the inequations in $\mathsf{System}_{\Sigma_{\mathsf{PSI}}}$ are satisfied when $\mathsf{VIS}, \mathsf{AR}, \mathsf{AntiVIS}$ are substituted for X_V, X_A, X_N, respectively.

Inequation (V1): $\mathsf{WR} \cup \mathsf{VIS}$. This is true, because $\mathsf{WR} \subseteq (\mathsf{WR} \cup \mathsf{WW})^+ = \mathsf{VIS}$,
Inequation (V3): $\mathsf{WW} \subseteq \mathsf{VIS}$. This can be proved as above: $\mathsf{WW} \subseteq (\mathsf{WR} \cup \mathsf{WW})^+ \subseteq \mathsf{VIS}$,
Inequation (V2): VIS ; $\mathsf{VIS} \subseteq \mathsf{VIS}$. This is trivially satisfied: VIS ; $\mathsf{VIS} = (\mathsf{WR} \cup \mathsf{WW})^+$; $(\mathsf{WR} \cup \mathsf{WW})^+ \subseteq (\mathsf{WR} \cup \mathsf{WW})^+ = \mathsf{VIS}$,
Inequation (A1): $\mathsf{WW} \subseteq \mathsf{AR}$. We have already proved that $\mathsf{WW} \subseteq \mathsf{VIS}$, hence it suffices to show that $\mathsf{VIS} \subseteq \mathsf{AR}$; this is done below,
Inequation (A2): $\mathsf{VIS} \subseteq \mathsf{AR}$. We have that

$$\mathsf{VIS} \subseteq \mathsf{VIS} \cup \bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] ; \mathsf{VIS}? ; \mathsf{RW}(x))^+ = \mathsf{AR},$$

Inequation (A4): AR ; $\mathsf{AR} \subseteq \mathsf{AR}$. We have that

$$\mathsf{AR} ; \mathsf{AR} = \left(\mathsf{VIS} \cup \bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] ; \mathsf{VIS}? ; \mathsf{RW}(x))^+\right) ; \left(\mathsf{VIS} \cup \bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] ; \mathsf{VIS}? ; \mathsf{RW}(x))^+\right) \overset{(62)}{\subseteq} (\mathsf{VIS} \cup \mathsf{WW}) ; (\mathsf{VIS} \cup \mathsf{WW}) \overset{(A1)}{=} \mathsf{VIS} ; \mathsf{VIS} \overset{(A2)}{\subseteq} \mathsf{AR}$$

Inequation (A3): $\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x]$; VIS ; $\mathsf{RW}(x) \subseteq \mathsf{AR}$. This inequation is trivially satisfied by the definition of AR:

$$\bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x] ; \mathsf{VIS} ; \mathsf{RW}(x) \subseteq \bigcup_{x \in \mathsf{Obj}} [\mathsf{Writes}_x] ; \mathsf{VIS}? ; \mathsf{RW}(x) \subseteq \mathsf{AR}$$

Inequation (N1): $\mathsf{RW} \subseteq \mathsf{AntiVIS}$. We have that $\mathsf{RW} \subseteq \mathsf{VIS}?$; RW ; $\mathsf{VIS}? = \mathsf{AntiVIS}$,
Inequation (N2): $\mathsf{VIS}?$; $\mathsf{RW} \subseteq \mathsf{AntiVIS}$: we have that $\mathsf{VIS}?$; $\mathsf{RW} \subseteq \mathsf{VIS}?$; RW ; $\mathsf{VIS}? = \mathsf{AntiVIS}$. Inequation (N3) can be proved similarly.

◂

Proof of Theorem 11(2). Let $\Delta_{\mathsf{PSI}} = \{\delta_{\mathsf{PSI}_0}\} \cup \{\delta_{\mathsf{PSI}(x)}\}_{x \in \mathsf{Obj}}$. Recall that

$$\delta_{\mathsf{PSI}_0} : \mathcal{G} \mapsto (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^+$$
$$\delta_{\mathsf{PSI}(x)} : \mathcal{G} \mapsto ((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^* ; \mathsf{RW}(x))^+.$$

We need to show that $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) = \mathsf{modelOf}(\Delta_{\mathsf{PSI}})$: for any execution $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$, $\mathsf{graph}(\mathcal{X}) \in \mathsf{Graphs}(\Delta_{\mathsf{PSI}})$, and for any $\mathcal{G} \in \mathsf{Graphs}(\Delta_{\mathsf{PSI}})$, there exists an execution $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$ such that $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$.

We prove this result in several step. First, define

$$\delta'_{\mathsf{PSI}} : \mathcal{G} \mapsto (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{+} \cup \bigcup_{x \in \mathsf{Obj}} ([\mathsf{Writes}_x] \ ; (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+} \ .$$

We prove that $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) = \mathsf{modelOf}(\{\delta'_{\mathsf{PSI}}\})$. By Theorem 14 we have that, for any $\mathcal{X} \in \mathsf{Executions}(\mathsf{PSI})$, the relation $\delta'_{\mathsf{PSI}}(\mathcal{G})$ is irreflexive, hence $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) \subseteq \mathsf{modelOf}(\{\delta'_{\mathsf{PSI}}\})$. Let then $\mathcal{G} \in \mathsf{modelOf}(\delta'_{\mathsf{PSI}})$, that is the relation $\delta'_{\mathsf{PSI}}(\mathcal{G})$ is irreflexive. By Proposition 59 we have that $(X_V = \_, X_A = \delta'_{\mathsf{PSI}}(\mathcal{G}), X_N = \_)$ is a solution to $\mathsf{System}_{\mathsf{PSI}}(\mathcal{G})$, and by Theorem 17 it follows that there exists a relation $\mathcal{X} \in \mathsf{Executions}(\Sigma_{\mathsf{PSI}})$ such that $\mathsf{graph}(\mathcal{X}) = \mathcal{G}$. That is, $\mathsf{modelOf}(\{\delta'_{\mathsf{PSI}}\}) \subseteq \mathsf{modelOf}(\Sigma_{\mathsf{PSI}})$.

Next, for any object $x \in \mathsf{Obj}$, define $\delta'_{\mathsf{PSI}(x)}(\mathcal{G}) = ([\mathsf{Writes}_x] \ ; (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+}$. It is immediate to observe that $\mathsf{modelOf}(\{\delta'_{\mathsf{PSI}}\}) = \mathsf{modelOf}(\{\delta_{\mathsf{PSI}_0}\} \cup \{\delta'_{\mathsf{PSI}(x)} \mid x \in \mathsf{Obj}\})$. In fact, for any $\mathcal{G} \in \mathsf{Graphs}$, we have that $\delta'_{\mathsf{PSI}}(\mathcal{G}) = \delta_{\mathsf{PSI}_0}(\mathcal{G}) \cup \bigcup_{x \in \mathsf{Obj}} \delta'_{\mathsf{PSI}(x)}(\mathcal{G})$, hence $\delta'_{\mathsf{PSI}}(\mathcal{G}) \cap \mathsf{Id} = \emptyset$ if and only if $\delta_{\mathsf{PSI}_0}(\mathcal{G}) \cap \mathsf{Id} = \emptyset$, and $\delta'_{\mathsf{PSI}}(x)(\mathcal{G}) \cap \mathsf{Id} = \emptyset$. At this point we have that $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) = \mathsf{modelOf}(\{\delta'_{\mathsf{PSI}}\}) = \mathsf{modelOf}(\{\delta_{\mathsf{PSI}_0}\} \cup \{\delta'_{\mathsf{PSI}(x) | x \in \mathsf{Obj}}\})$.

As a last step, we show that for each dependency graph $\mathcal{G}$ and object x, the relation $\delta'_{\mathsf{PSI}(x)}(\mathcal{G})$ is irreflexive if and only if the relation $\delta_{\mathsf{PSI}(x)}(\mathcal{G})$ is irreflexive, where we recall that $\delta_{\mathsf{PSI}(x)}(\mathcal{G}) = ((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+}$. An immediate consequence of this fact is that $\mathsf{modelOf}(\Sigma_{\mathsf{PSI}}) = \mathsf{modelOf}(\{\delta_{\mathsf{PSI}_0}\} \cup \{\delta_{\mathsf{PSI}(x)} \mid x \in \mathsf{Obj}\}) = \mathsf{modelOf}(\Delta_{\mathsf{PSI}})$, which is exactly what we want to prove.

Note that $\delta'_{\mathsf{PSI}(x)}(\mathcal{G}) = ([\mathsf{Writes}_x] \ ; (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+} \subseteq)(\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+} = \delta_{\mathsf{PSI}(x)}(\mathcal{G})$: if $\delta_{\mathsf{PSI}(x)}(\mathcal{G})$ is irreflexive, then so if $\delta'_{\mathsf{PSI}(x)}(\mathcal{G})$. Finally, suppose that $\delta'_{\mathsf{PSI}(x)}(\mathcal{G}) \cap \mathsf{Id} \subseteq \emptyset$. That is, $([\mathsf{Writes}_x] \ ; (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x))^{+} \cap \mathsf{Id} \subseteq \emptyset$. We apply the following Theorem from Kleene Algebra: for any relations $R_1, R_2 \subseteq \mathcal{T}_{\mathcal{G}} \times \mathcal{T}_{\mathcal{G}}$, $(R_1 \ ; R_2)^{+} = R_1 \ ; (R_2 \ ; R_1)^{*} \ ; R_2$. This leads to the following:

$$([\mathsf{Writes}_x] \ ; ((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}(x) \ ; [\mathsf{Writes}_x])^{*} \ ; ((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}(x))) \cap \mathsf{Id} \subseteq \emptyset$$

Also, by Proposition 35, the latter can be rewritten as follows:

$$(((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x) \ ; [\mathsf{Writes}_x])^{*} \ ; (\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x) \ ; [\mathsf{Writes}_x])) \cap \mathsf{Id} \subseteq \emptyset$$

which can be simplified into

$$((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}_{\mathcal{G}}(x) \ ; [\mathsf{Writes}_x])^{+} \cap \mathsf{Id} \subseteq \emptyset.$$

As a last step, note that $\mathsf{RW}_{\mathcal{G}}(x) \ ; [\mathsf{Writes}_x] \subseteq \mathsf{RW}_{\mathcal{G}}(x)$, hence we have

$$((\mathsf{WR}_{\mathcal{G}} \cup \mathsf{WW}_{\mathcal{G}})^{*} \ ; \mathsf{RW}(x))^{+} \cap \mathsf{Id} \subseteq \emptyset$$

which is exactly $\delta_{\mathsf{PSI}(x)}(\mathcal{G}) \cap \mathsf{Id} \subseteq \emptyset$. ◂