**Made available by Hasselt University Library in https://documentserver.uhasselt.be**

# A Declarative Semantics for Dedalus

Peer-reviewed author version

Alvaro, Peter; AMELOOT, Tom; Hellerstein, Joseph M.; Marczak, William R. & VAN DEN BUSSCHE, Jan (2013) A Declarative Semantics for Dedalus.

Handle: http://hdl.handle.net/1942/14572

# A Declarative Semantics for Dedalus

Peter Alvaro, Tom J. Ameloot, Joseph M. Hellerstein,
William R. Marczak, Jan Van den Bussche

**Abstract**

The language Dedalus is a Datalog-like language in which distributed computations and networking protocols can be programmed, in the spirit of the Declarative Networking paradigm. Whereas recently formal operational semantics for Dedalus-like languages have been developed, a purely declarative semantics has been lacking so far. The challenge is to capture precisely the amount of nondeterminism that is inherent to distributed computations due to concurrency, networking delays, and asynchronous communication. This paper shows how a declarative semantics can be obtained by simply using the well-known stable model semantics for Datalog with negation. This semantics is applied to the Dedalus rules after they are modified to account for nondeterministic arrival times of messages, and augmented with control rules which model causality. The main result then is that, as far as fair operational runs are concerned, the proposed declarative semantics matches exactly the previously proposed formal operational semantics.

## 1 Introduction

In recent years, logic programming has been proposed as an attractive foundation for distributed and cloud programming, building on work in declarative networking [26]. In recent years we are also seeing a more general resurgence of interest in Datalog, e.g., [12, 20]. A notable promising property of Datalog is that it allows complex distributed algorithms and protocols to be expressed in relatively few lines of code [21, 19]. One of the latest languages proposed in declarative networking is Dedalus [5, 6, 19], a Datalog-inspired language that has influenced other recent language designs for distributed and cloud computing such as Webdamlog [1] and Bloom [4]. Moreover, issues related to Dedalus and data-oriented distributed computing are recently receiving some attention at database theory conferences [18, 7, 1, 8, 32].

It is well understood how an *operational* semantics for Dedalus-like languages can be defined formally [13, 28, 17, 7]. Such a formal semantics is typically defined as a transition system. The transition system is infinite even if the distributed computation is working on a finite input database, because compute nodes can run indefinitely; moreover, they can keep on sending messages so that

an unbounded number of messages can be floating around in the network.[1] In addition, the transition system is highly nondeterministic, because nodes work concurrently, communication is asynchronous, and messages can be delayed and eventually be delivered out of order by the network.

On the other hand, it remains unclear how (and if) a purely *declarative* formal semantics can be given for the languages used in declarative (!) networking. This has been lacking so far, and the purpose of this paper is to contribute towards filling this gap. Concretely, our work can be summarized as follows.

1. We begin by presenting a formal operational semantics for Dedalus. As mentioned above, this part is quite standard. Our definition leads to the notion of fair runs of a Dedalus program $\mathcal{P}$ on an input distributed database instance $H$. Runs represent distributed computations and, due to the nondeterminism mentioned above, there are typically many fair runs of $\mathcal{P}$ on $H$.

2. Each run respects a causal order (which is a partial order) that relates the local steps of the different compute nodes through chains of local steps and communicated messages. This order indicates what events "happened before" which other events [11]. Now, the computation of each run can be described by a structure which we call a trace, which includes for each compute node in the network the detailed information about the local steps it has performed and about the messages it has sent and received. The trace conforms to the causal order.

3. The main idea now is that the traces of runs can be obtained precisely as the set of *stable models* [15] of $\mathcal{P}$ on $H$. A few manipulations are needed before we can aim for such a result, however, because the Dedalus program $\mathcal{P}$ is not really a Datalog$^\neg$ program. Indeed, the language Dedalus provides special "inductive rules" and "asynchronous rules" that are used for persisting memory across local computation steps and sending messages respectively. First, we will transform these rules into Datalog$^\neg$ rules that simulate their effect, where asynchronous rules will nondeterministically choose the arrival times of messages [22, 30]. Furthermore, $\mathcal{P}$ is augmented with a fixed, instance- and network-independent set of rules that express causality on the messages. Applying the stable model semantics to such transformed Dedalus programs constitutes the declarative semantics.

We believe that our result is interesting because it shows the equivalence between two quite different ways to define the semantics of a Dedalus program.

Perhaps most importantly, the result is of interest for grounding a representative database language for distributed and cloud computing in a well-motivated model-theoretic semantics. Indeed, our characterization provides a purely declarative axiomatization of fair distributed program behaviors in terms

[1] We use the common term "node" as a synonym for an individual "computer" in a network or cluster.

of the stable models of a logical theory (finite set of Datalog$^\neg$ rules). Specifically, we have succeeded in capturing in Datalog$^\neg$ the operational notion of causality, which focuses on just the key features of the distributed computation: the state of each compute node at each local time, and at what local times the nodes send and receive messages. Hence our declarative semantics reasons from the perspective of the local times of each node, which is a justified approach since there is no common "global clock" in a distributed environment [11].

It should be noted that the declarative semantics studied in the current work perhaps can not directly be used in practical applications, because in the semantics there is a reference to an infinite number of local computation steps of the nodes. But we hope our work can provide insights for the design of a more intuitive declarative semantics, used by developers of distributed applications.

As mentioned, many Datalog-inspired languages have been proposed to implement distributed applications [26, 28, 17, 1], and they contain several features such as aggregation and non-determinism (choice), that result in powerful languages. But the essential features that all these languages possess, are reasoning about distributed state and representing message passing. We think of the language Dedalus, as we define it here, as a minimalistic extension of Datalog to provide just these essential features. For this reason, we expect that the current work can serve as a theoretical base that can be extended to more powerful language features as well.

An area of artificial intelligence that is closely related to declarative networking is that of programming multi-agent systems in declarative languages. The knowledge of an agent can be expressed by a logic program, which also allows for non-monotone reasoning, and agents update their knowledge by modifying the rules in these logic programs [25, 29, 24]. The language LUPS [3] was designed to specify such dynamic updates to logic programs, and LUPS is also a declarative language itself. After applying a sequence of updates specified in LUPS, the semantics of the resulting logic program can be defined in an inductive way. But an interesting connection to this current work, is that the semantics can also be given by first syntactically translating the original program and its updates into a single normal logic program, after which the stable model semantics is applied [3]. This has also lead to a practical implementation of LUPS. It should be noted however that in this second semantics, there is no modeling of causality or the sending of messages.

This paper is organized as follows. In Section 2 we give preliminaries, including the language Dedalus. In Section 3 we give the operational semantics for Dedalus and some example programs. In Section 4 we define the declarative semantics for Dedalus. In Section 5 we give our main result that relates the operational and declarative semantics. The proof of this result is given in two parts, in Sections 6 and 7. We finish with the conclusion in Section 8.

# 2 Preliminaries

## 2.1 Databases and Facts

A *database schema* $\mathcal{D}$ is a set of pairs $(R, k)$ where $R$ is a *relation name* and $k \in \mathbb{N}$ its associated *arity*. A relation name occurs at most once in a database schema. We often write $(R, k)$ as $R^{(k)}$.

We assume some infinite universe **dom** of atomic data values. A *fact* $\boldsymbol{f}$ is a pair $(R, \bar{a})$, often denoted as $R(\bar{a})$, where $R$ is a relation name and $\bar{a}$ is a tuple of values over **dom**. For a fact $R(\bar{a})$, we call $R$ the *predicate*. We say that a fact $R(a_1, \ldots, a_k)$ is *over* a database schema $\mathcal{D}$ if $R^{(k)} \in \mathcal{D}$. A database *instance* $I$ over a database schema $\mathcal{D}$ is a set of facts over $\mathcal{D}$. For a database schema $\mathcal{D}' \subseteq \mathcal{D}$, we write $I|_{\mathcal{D}'}$ to denote the subset of facts in $I$ that are over $\mathcal{D}'$.

For a set of facts $I$, we write $adom(I)$ to denote the set of all values that occur in the facts of $I$.

## 2.2 Datalog with Negation

Below we recall the language Datalog with negation [2], which we abbreviate as Datalog$^\neg$.

Let **var** be a universe of *variables*, disjoint from **dom**. We will use typewriter font for variables, to better distinguish them from values in **dom**. An *atom* is of the form $R(u_1, \ldots, u_k)$ where $R$ is a relation name and $u_i \in \mathbf{var} \cup \mathbf{dom}$ for $i = 1, \ldots, k$. We call $R$ the *predicate*. If an atom contains no data values, then we call it *constant-free*. A *literal* is an atom or a negated atom. A literal that is an atom is called *positive* and otherwise it is called *negative*.

A Datalog$^\neg$ *rule* $\varphi$ is a triple

$$(head_\varphi,\ pos_\varphi,\ neg_\varphi)$$

where $head_\varphi$ is an atom, and $pos_\varphi$ and $neg_\varphi$ are sets of atoms. The components $head_\varphi$, $pos_\varphi$ and $neg_\varphi$ are called respectively the *head*, the *positive body atoms* and the *negative body atoms*. The union of the last two sets is called the *body atoms*. Note that in our formalization, the set $neg_\varphi$ contains just atoms, not negative literals. Every Datalog$^\neg$ rule $\varphi$ must have a head, whereas $pos_\varphi$ and $neg_\varphi$ may be empty. If $neg_\varphi = \emptyset$ then $\varphi$ is called *positive*. If all atoms comprising $\varphi$ are constant-free, then $\varphi$ is called *constant-free*.

A rule $\varphi$ may be written in the conventional syntax. For instance, if $head_\varphi = T(\mathtt{u}, \mathtt{v})$, $pos_\varphi = \{R(\mathtt{u}, \mathtt{v})\}$ and $neg_\varphi = \{S(\mathtt{v})\}$, with $\mathtt{u}, \mathtt{v} \in \mathbf{var}$, then we can write $\varphi$ as

$$T(\mathtt{u}, \mathtt{v}) \leftarrow R(\mathtt{u}, \mathtt{v}),\ \neg S(\mathtt{v}).$$

The specific ordering of literals to the right of the arrow is arbitrary.

We call $\varphi$ *safe* if the variables occurring in $head_\varphi$ and $neg_\varphi$ also occur in $pos_\varphi$. The set of variables of $\varphi$ is denoted $vars(\varphi)$. If $vars(\varphi) = \emptyset$ then $\varphi$ is called *ground*, in which case we will consider $\{head_\varphi\} \cup pos_\varphi \cup neg_\varphi$ to be a set of facts.

Let $\mathcal{D}$ be a database schema. A rule $\varphi$ is said to be *over schema* $\mathcal{D}$ if for each atom $R(u_1, \ldots, u_k) \in \{head_\varphi\} \cup pos_\varphi \cup neg_\varphi$ we have $R^{(k)} \in \mathcal{D}$. A Datalog$^\neg$ *program* $P$ *over* $\mathcal{D}$ is a set of safe Datalog$^\neg$ rules over $\mathcal{D}$. We call $P$ *constant-free* if all rules in $P$ are constant-free. We will write $sch(P)$ to denote the database schema that $P$ is over. We define $idb(P) \subseteq sch(P)$ to be the database schema consisting of all relations occurring in rule-heads of $P$. We abbreviate $edb(P) = sch(P) \setminus idb(P)$.[2]

An *input* for $P$ is a database instance over $sch(P)$. Note that we allow inputs to already contain facts over $idb(P)$, and the reason for this slightly unconventional input definition will become clear in Section 3.

Let $P$ be a Datalog$^\neg$ program. Let $I$ be an instance over $sch(P)$. Let $\varphi \in P$. A *valuation for* $\varphi$ is a total function $V : vars(\varphi) \to \mathbf{dom}$. Note that there is only one valuation for ground rules, namely, the one having an empty domain. Now, we define the *application* of $V$ to an atom $R(u_1, \ldots, u_k)$ of $\varphi$, denoted $V(R(u_1, \ldots, u_k))$, as the *fact* $R(a_1, \ldots, a_k)$ where for $i = 1, \ldots, k$ we have $a_i = V(u_i)$ if $u_i \in \mathbf{var}$ and $a_i = u_i$ otherwise. In words: the application of valuation $V$ replaces the variables by data values and leaves the old data values unchanged. This notation is naturally extended to a set of atoms, which results in a set of facts. Now, the valuation $V$ is said to be *satisfying for* $\varphi$ *on* $I$ if $V(pos_\varphi) \subseteq I$, $V(neg_\varphi) \cap I = \emptyset$. If this is so, then $\varphi$ is said to *derive* the fact $V(head_\varphi)$.

Remark: note that our definition of Datalog$^\neg$ is general enough to allow for the simulation of nonequalities like $u \neq v$ in rule bodies, because a Datalog$^\neg$ program can compute a binary *idb*-relation `equal` in which identical tuples like $(a, a)$ appear, and then negation can be applied to this relation in rule bodies.

### 2.2.1 Positive and Semi-positive

Let $P$ be a Datalog$^\neg$ program. We say that $P$ is *positive* if it has only positive rules. We say that $P$ is *semi-positive* if for each rule $\varphi \in P$, all predicates used in atoms of $neg_\varphi$ are contained in $edb(P)$. Naturally, positive programs are semi-positive.

Let $P$ be a semi-positive Datalog$^\neg$ program. We now give the semantics of $P$ [2]. We define the *immediate consequence operator* $T_P$ that maps each instance $J$ over $sch(P)$ to the instance $J' = J \cup A$ where $A$ is the set of facts derived by all possible satisfying valuations for the rules of $P$ on $J$. Note that $adom(J') \subseteq adom(J)$.

Let $I$ be an instance over $sch(P)$. Consider the infinite sequence $I_0$, $I_1$, $I_2$, etc, that is inductively defined as follows: $I_0 = I$ and $I_i = T_P(I_{i-1})$ for each $i \geq 1$. We define the *output of* $P$ *on input* $I$, denoted $P(I)$, as $\bigcup_j I_j$; this is the *minimal fixpoint* of the $T_P$ operator. Note that $I \subseteq P(I)$. When $I$ is finite, the fixpoint is finite and can be computed in polynomial time.

[2]The abbreviation "idb" stands for "intensional database schema" and "edb" stands for "extensional database schema" [2].

### 2.2.2 Stratified Semantics

We now recall the notion of stratified semantics for a Datalog$^\neg$ program [2]. Let $P$ be a Datalog$^\neg$ program. To improve readability, as a slight abuse of notation, here we will treat $idb(P)$ as a set of only relation names (without associated arities). The program $P$ is called *syntactically stratifiable* if there is a function $\sigma : idb(P) \to \{1, \ldots, |idb(P)|\}$ such that for each rule $\varphi \in P$, having some head predicate $T$, the following conditions are satisfied:

- $\sigma(R) \leq \sigma(T)$ for each $R(\bar{v}) \in pos_\varphi|_{idb(P)}$;
- $\sigma(R) < \sigma(T)$ for each $R(\bar{v}) \in neg_\varphi|_{idb(P)}$.

For $R \in idb(P)$, we call $\sigma(R)$ the *stratum number* of $R$. For technical convenience, we may assume that if there is an $R \in idb(P)$ with $\sigma(R) > 1$ then there is an $S \in idb(P)$ with $\sigma(S) = \sigma(R) - 1$.

Intuitively, the function $\sigma$ partitions $P$ into a sequence of semi-positive Datalog$^\neg$ programs $P_1, \ldots, P_k$ with $k \leq |idb(P)|$ such that for each $i = 1, \ldots, k$, the program $P_i$ is the set of rules of $P$ whose head predicate has stratum number $i$. Rules with the same head predicate are always in the same semi-positive program. This sequence of semi-positive programs is called a *syntactic stratification* of $P$. We can now apply the *stratified semantics* to $P$: for an input $I$ over $sch(P)$, we first compute the fixpoint $P_1(I)$, then the fixpoint $P_2(P_1(I))$, etc. The *output of $P$ on input $I$*, denoted $P(I)$, is then defined as $P_k(P_{k-1}(\ldots P_1(I)\ldots))$. It is well known that the output of $P$ does not depend on the chosen syntactic stratification (in the case that more than one exists).

Not all Datalog$^\neg$ programs are syntactically stratifiable.

### 2.2.3 Stable Model Semantics

We now recall the notion of stable model semantics [15, 30]. Let $P$ be a Datalog$^\neg$ program and let $I$ be a database instance over $sch(P)$. Let $\varphi \in P$. Let $V$ be a valuation for $\varphi$ whose image is contained in $adom(I)$. Valuation $V$ does not have to be satisfying for $\varphi$ on $I$. Together, $V$ and $\varphi$ give rise to a ground rule $\psi$, that is precisely $\varphi$ except that each $u \in vars(\varphi)$ is replaced by $V(u)$. We call $\psi$ a *ground rule of $\varphi$ with respect to $I$*. Let $ground(\varphi, I)$ denote the set of all ground rules of $\varphi$ that we can make with respect to $I$. The *ground program of $P$ on input $I$*, denoted $ground(P, I)$, is defined as $\bigcup_{\varphi \in P} ground(\varphi, I)$.

Let $M$ be a set of facts over the schema $sch(P)$. We write $ground_M(P, I)$ to denote the program obtained from $ground(P, I)$ as follows:

1. remove every rule $\psi \in ground(P, I)$ for which $neg_\psi \cap M \neq \emptyset$;
2. remove the negative (ground) body atoms from all remaining rules.

Note that $ground_M(P, I)$ is a positive program. We say that $M$ is a *stable model of $P$ on input $I$* if $M$ is the output of $ground_M(P, I)$ on input $I$. This implies that $I \subseteq M$ and $adom(M) \subseteq adom(I)$ by the semantics of positive Datalog$^\neg$ programs.

Not all Datalog$^\neg$ programs have stable models on every input.

## 2.3 Network and Distributed Databases

A *(computer) network* is a nonempty finite set $\mathcal{N}$ of *nodes*, which are values in **dom**. Intuitively, $\mathcal{N}$ represents a set of identifiers of compute nodes involved in a distributed system. We do not explicitly represent communication channels (edges) because we will work in a model where any node $x$ can send a message to any other node $y$, as long as $x$ knows about $y$. This knowledge can come from local input relations of $x$ or from messages that $x$ has received itself. When using Dedalus for general distributed or cluster computing, the delivery of messages is handled by the network layer, which is abstracted away. But Dedalus programs can also be used to describe the network layer itself [26, 19]. In that case, we would restrict attention to programs where nodes only send messages to nodes to which they are explicitly linked; these nodes would again be provided as input.

A *distributed database instance* $H$ over a network $\mathcal{N}$ and a database schema $\mathcal{D}$ is a function that maps every node of $\mathcal{N}$ to an ordinary finite database instance over $\mathcal{D}$. This represents how data over the same schema $\mathcal{D}$ is spread over the nodes of a network.

## 2.4 Dedalus Programs

We now recall the language Dedalus, that can be used to describe distributed computations [5, 6, 19]. Essentially, Dedalus is an extension of Datalog$^\neg$ to represent updatable memory for the nodes of a network and to provide a mechanism for communication between these nodes. In the current work, we present Dedalus as Datalog$^\neg$ extended with a simple annotation mechanism, which keeps the notations simpler.[3]

Let $\mathcal{D}$ be a database schema. We write $\mathbf{B}\{\bar{\mathtt{v}}\}$, where $\bar{\mathtt{v}}$ is a tuple of variables, to denote any sequence $\beta$ of literals over database schema $\mathcal{D}$, such that the variables in $\beta$ are precisely those in the tuple $\bar{\mathtt{v}}$. Let $R(\bar{\mathtt{u}})$ denote any atom over $\mathcal{D}$. There are three types of Dedalus rules over $\mathcal{D}$:

- A *deductive* rule is a normal Datalog$^\neg$ rule over $\mathcal{D}$.
- An *inductive* rule is of the form

$$R(\bar{\mathtt{u}})\bullet \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}\}.$$

- An *asynchronous* rule is of the form

$$R(\bar{\mathtt{u}}) \mid \mathtt{y} \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}, \mathtt{y}\}.$$

For inductive rules, the annotation '$\bullet$' can be likened to the transfer of "tokens" in a Petri net from the old state to the new state. For asynchronous rules, the annotation '$\mid \mathtt{y}$' with $\mathtt{y} \in \mathbf{var}$ means that the derived head facts are transferred ("piped") to the node represented by $\mathtt{y}$. Deductive, inductive and asynchronous

[3] These annotations correspond to syntactic sugar in the previous presentations of Dedalus.

rules will express respectively local computation, updatable memory, and message sending (cf. Section 3). Like in Section 2.2, a Dedalus rule is called *safe* if all its variables occur in at least one positive body atom.

To illustrate, if $\mathcal{D} = \{R^{(2)}, S^{(1)}, T^{(2)}\}$, then the following three rules are examples of, respectively, deductive, inductive and asynchronous rules over $\mathcal{D}$:

$$T(\mathtt{u}, \mathtt{v}) \leftarrow R(\mathtt{u}, \mathtt{v}),\ \neg S(\mathtt{v}).$$

$$T(\mathtt{u}, \mathtt{v})\bullet \leftarrow R(\mathtt{u}, \mathtt{v}).$$

$$T(\mathtt{u}, \mathtt{v}) \mid \mathtt{y} \leftarrow R(\mathtt{u}, \mathtt{v}),\ S(\mathtt{y}).$$

Now consider the following definition:

**Definition 2.1.** A Dedalus *program over a schema* $\mathcal{D}$ is a set of deductive, inductive and asynchronous Dedalus rules over $\mathcal{D}$, such that all rules are safe, and the set of deductive rules is syntactically stratifiable.

In the current work, we will additionally assume that Dedalus programs are constant-free, as is common in the theory of database query languages, and which is not really a limitation, since constants that are important for the program can always be indicated by unary relations in the input.

Let $\mathcal{P}$ be a Dedalus program. We write $sch(\mathcal{P})$ to denote the schema that $\mathcal{P}$ is over. Because $\mathcal{P}$ is a set of rules, the definitions of $idb(\mathcal{P})$ and $edb(\mathcal{P})$ are like for Datalog$^\neg$ programs.

An *input* for $\mathcal{P}$ is a distributed database instance $H$ over some network $\mathcal{N}$ and the schema $edb(\mathcal{P})$.

# 3 Operational Semantics

In this section we give an operational semantics for Dedalus. We describe how a network executes a Dedalus program $\mathcal{P}$ when an input distributed database is given. This operational semantics is in line with earlier formal work on declarative networking [13, 28, 17, 7, 1].

The essence of the operational semantics is as follows. By definition, the input distributed database is over a certain network. Every node of the network runs the *same* Dedalus program, and a node has access only to its own local state and any received messages. The nodes are made active one by one in some arbitrary order, and this continues an infinite number of times. During each active moment of a node $x$, called a *local (computation) step*, node $x$ receives message facts and applies its deductive, inductive and asynchronous rules. Concretely, the deductive rules, forming a stratified Datalog$^\neg$ subprogram, are applied to the incoming messages and the previous state of $x$. Next, the inductive rules are applied to the output of the deductive subprogram, and these allow $x$ to store facts in its memory: these facts become visible in the next local step of $x$. Finally, the asynchronous rules are also applied to the output of the deductive subprogram, and these allow $x$ to send facts to the other nodes or to itself.

These facts become visible at the addressee after some arbitrary delay, which represents asynchronous communication, as occurs for instance on the Internet. We assume that all messages are eventually delivered (and are thus never lost). We will refer to local steps simply as "steps".

In the next subsections, we make the above sketch more concrete. We will start by giving some illustrative example programs.

## 3.1 Example Programs

In this section we give some example Dedalus programs that help illustrate the language. For the purposes of this section, we assume that every node always has at least the local unary input relations `Id` and `Node`, that contain respectively the identifier of the local node and the identifiers of all nodes (including the local node). Additional input relations will use a different name, and for the sake of simplicity, we will assume that the relations `Id` and `Node` are automatically initialized correctly when we define the inputs for the Dedalus programs below.

We also briefly mention that it is possible to define the output of Dedalus programs based on so-called *ultimate facts*, which are the facts on every node that are eventually derived (by the deductive subprogram) during every step [27]. The examples below follow this principle.

**Example 3.1.** Suppose that each node has a binary input relation $R$ that represents a graph. We want to compute at each node the transitive closure of the global graph that is obtained by uniting the local input graphs of all nodes. This output should be stored in a relation $T^{(2)}$ at each node.

For any network $\mathcal{N}$, for any distributed database instance over $\mathcal{N}$ and relation $R^{(2)}$, the following Dedalus program computes the required output at each node, in a well-known way [26]:

$$T(\mathtt{u},\mathtt{v}) \mid \mathtt{y} \leftarrow R(\mathtt{u},\mathtt{v}),\ \mathtt{Node}(\mathtt{y}).$$
$$T(\mathtt{u},\mathtt{v}) \mid \mathtt{y} \leftarrow R(\mathtt{u},\mathtt{w}),\ T(\mathtt{w},\mathtt{v}),\ \mathtt{Node}(\mathtt{y}).$$
$$T(\mathtt{u},\mathtt{v})\bullet \leftarrow T(\mathtt{u},\mathtt{v}).$$

The first asynchronous rule lets each node broadcast all of its local input $R$-facts as $T$-facts to every node, including itself. The second asynchronous rule lets each node take an incoming $T$-fact, join it with local $R$-facts, and broadcast the resulting transitive edges again to every node. The last rule is inductive, and it continuously persists all received $T$-facts to the next step, so that the relation $T$ steadily grows at each node. After a while, every node will have accumulated all original graph edges and the transitive edges in relation $T$. □

The previous example showed how a recursive, monotone computation can be expressed. The next example illustrates how Dedalus can also be used to do a nonmonotone computation in a distributed setting.

**Example 3.2.** In this example, nullary relations are used as booleans: *true* is represented by the nonempty relation and *false* by the empty relation. Suppose

that each node has a nullary input relation $S$. We want to compute at each node a nullary fact $T(\,)$ if and only if the relation $S$ is empty at *all* nodes. This is a nonmonotone computation. Indeed, if all nodes have an empty relation $S$ then we produce $T(\,)$ on all nodes, and if at least one node has a nonempty relation $S$ then we should not output $T(\,)$ on any node.

For every network $\mathcal{N}$, the following Dedalus program computes the desired output at each node:

$$\texttt{empty(x)} \mid \texttt{y} \leftarrow \texttt{Id(x)}, \neg S(\,), \texttt{Node(y)}.$$
$$\texttt{empty(x)}\bullet \leftarrow \texttt{empty(x)}.$$
$$\texttt{missing}(\,) \leftarrow \texttt{Node(x)}, \neg\texttt{empty(x)}.$$
$$T(\,) \leftarrow \neg\texttt{missing}(\,).$$

The first asynchronous rule lets a node broadcast its own identifier to every node (including itself) if its local input relation $S$ is empty. The second rule is inductive and it persists the received node identifiers. The third rule is deductive, and it checks whether the identifiers of all nodes have been received (`missing` is false) or not (`missing` is true). Because the rule is deductive, the relation `missing` is recomputed during every local step of a node. The last rule, which is also deductive, produces a $T(\,)$-fact at every node that has received all node identifiers.

If indeed every node has an empty $S$-relation, after a while, all nodes have received all node identifiers, and from that moment onwards, all nodes produce $T(\,)$ in every step. In the other case, when at least one of the nodes does not have an empty $S$-relation, no node will receive the identifier of that node and thus no node will ever produce $T(\,)$. □

We present the formal operational semantics in the next subsections.

## 3.2 Configurations

A configuration describes the network at a certain point in its evolution. Let $H$ be an input distributed database instance for $\mathcal{P}$, over a network $\mathcal{N}$. We define a *configuration $\rho$ of $\mathcal{P}$ on $H$* to be a pair $(\mathrm{st}^\rho, \mathrm{bf}^\rho)$ where

- $\mathrm{st}^\rho$ is a function that maps each node of $\mathcal{N}$ to a set of facts over $sch(\mathcal{P})$;
- $\mathrm{bf}^\rho$ is a function that maps each node of $\mathcal{N}$ to a set of pairs of the form $\langle i, \boldsymbol{f}\rangle$, where $i \in \mathbb{N}$ and $\boldsymbol{f}$ is a fact over $idb(\mathcal{P})$.

We call $\mathrm{st}^\rho$ the *state* and $\mathrm{bf}^\rho$ the *(message) buffer* respectively. The state $\mathrm{st}^\rho$ says for each node what facts it has stored in its memory, and the message buffer $\mathrm{bf}^\rho$ says for each node what messages have been sent to it but that are not yet received. So, $\mathrm{bf}^\rho$ represents the contents of the transmission channels between the nodes. The reason for having numbers $i$, called *send-tags*, attached to facts in the image of $\mathrm{bf}^\rho$ is merely a technical convenience: these numbers help separate multiple instances of the same fact when it is sent at different

moments (to the same addressee), and these send-tags will not be visible to the Dedalus program.[4]

The *start configuration of* $\mathcal{P}$ *on input* $H$ is the configuration $\rho$ defined by

- $\mathrm{st}^\rho(x) = H(x)$ for each $x \in \mathcal{N}$;
- $\mathrm{bf}^\rho(x) = \emptyset$ for each $x \in \mathcal{N}$.

In words: for every node, the state is initialized with its local input fragment in $H$, and there are no sent messages. We denote this configuration as $\mathit{start}(\mathcal{P}, H)$.

### 3.3 Subprograms

We look at the operations that are executed locally during each step of a node. We have mentioned that the three types of Dedalus rules each have their own purpose in the operational semantics. For this reason, we split the program $\mathcal{P}$ into three subprograms, that contain respectively the deductive, inductive and asynchronous rules. In Section 3.4, we describe how these subprograms are used in the operational semantics.

- First, we define $\mathit{deduc}_\mathcal{P}$ to be the Datalog$^\neg$ program consisting of precisely all deductive rules of $\mathcal{P}$.
- Secondly, we define $\mathit{induc}_\mathcal{P}$ to be the Datalog$^\neg$ program consisting of all inductive rules of $\mathcal{P}$ after the annotation '$\bullet$' in their head is removed.
- Thirdly, we define $\mathit{async}_\mathcal{P}$ to be the Datalog$^\neg$ program consisting of precisely all rules
$$T(\mathtt{y}, \bar{\mathtt{u}}) \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \mathtt{y}\}$$
where
$$T(\bar{\mathtt{u}}) \mid \mathtt{y} \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \mathtt{y}\}$$
is an asynchronous rule of $\mathcal{P}$. So, we basically put the variable $\mathtt{y}$ as the first component in the (extended) head atom. The intuition for the generated head facts is that the first component will represent the addressee.

Note that the programs $\mathit{deduc}_\mathcal{P}$, $\mathit{induc}_\mathcal{P}$ and $\mathit{async}_\mathcal{P}$ are just Datalog$^\neg$ programs over the schema $\mathit{sch}(\mathcal{P})$, or a subschema thereof. Moreover, $\mathit{deduc}_\mathcal{P}$ is syntactically stratifiable because the deductive rules in every Dedalus program must be syntactically stratifiable. It is possible however that $\mathit{induc}_\mathcal{P}$ and $\mathit{async}_\mathcal{P}$ are not syntactically stratifiable. Now we define the semantics of each of these three subprograms.

Let $I$ be a database instance over $\mathit{sch}(\mathcal{P})$. During each step of a node, the intuition of the deductive rules is that they "complete" the available facts by adding all new facts that can be logically derived from them. This calls for a fixpoint semantics, and for this reason, we define the *output of* $\mathit{deduc}_\mathcal{P}$ *on input*

[4] We could have equivalently modeled message buffers with multisets, where messages are not tagged. The tags however are technically easier to work with.

$I$, denoted as $\mathit{deduc}_{\mathcal{P}}(I)$, to be given by the stratified semantics. This implies $I \subseteq \mathit{deduc}_{\mathcal{P}}(I)$. Importantly, $I$ is allowed to contain facts over $\mathit{idb}(\mathcal{P})$, and the intuition is that these facts were derived during a previous step (by inductive rules) or received as messages (as sent by asynchronous rules). This will become more explicit in Section 3.4.

During each step of a node, the intuition behind the inductive rules is that they store facts in the memory of the node, and these stored facts will become visible during the next step. There is no notion of a fixpoint here because facts that will become visible in the next step are not available in the current step to derive more facts. For this reason, we define the *output of* $\mathit{induc}_{\mathcal{P}}$ *on input* $I$ to be the set of facts derived by the rules of $\mathit{induc}_{\mathcal{P}}$ for all possible satisfying valuations in $I$, in just one derivation step. This output is denoted as $\mathit{induc}_{\mathcal{P}}\langle I\rangle$. Possibly $I \cap \mathit{induc}_{\mathcal{P}}\langle I\rangle = \emptyset$.

During each step of a node, the intuition behind the asynchronous rules is that they generate "message" facts that are to be sent around the network. The *output for* $\mathit{async}_{\mathcal{P}}$ *on input* $I$ is defined in the same way as for $\mathit{induc}_{\mathcal{P}}$, except that we now use the rules of $\mathit{async}_{\mathcal{P}}$ instead of $\mathit{induc}_{\mathcal{P}}$. This output is denoted as $\mathit{async}_{\mathcal{P}}\langle I\rangle$. The intuition for not requiring a fixpoint for $\mathit{async}_{\mathcal{P}}$ is that a message fact will arrive at another node, or at a later step of the sender node, and can therefore not be read during sending.

Regarding data complexity [31], for each subprogram the output can be computed in PTIME with respect to the size of its input.

## 3.4 Transitions and Runs

We will now define how we can go from one configuration $\rho_a$ to another configuration $\rho_b$. This is formalized by means of *transitions*, and they describe how one active node does a local computation step to update its state and to send messages around the network. Such transitions can be chained to form a *run* that describes a full execution of the Dedalus program on the given input.

As a small notational aid, for a set $m$ of pairs of the form $\langle i, \boldsymbol{f}\rangle$, we define $\mathit{untag}(m) = \{\boldsymbol{f} \mid \exists i \in \mathbb{N} : \langle i, \boldsymbol{f}\rangle \in m\}$.

A *(delivery) transition with send-tag* $i \in \mathbb{N}$ is a five-tuple $(\rho_a, x, m, i, \rho_b)$, also denoted as

$$\rho_a \xrightarrow[i]{x,\, m} \rho_b,$$

such that $\rho_a$ and $\rho_b$ are configurations of $\mathcal{P}$ on input $H$, $x \in \mathcal{N}$, $m \subseteq \mathrm{bf}^{\rho_a}(x)$, and, letting

$$\begin{aligned} I &= \mathrm{st}^{\rho_a}(x) \cup \mathit{untag}(m), \\ D &= \mathit{deduc}_{\mathcal{P}}(I), \\ \delta^{i\to y} &= \{\langle i,\, R(\bar{a})\rangle \mid R(y, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D\rangle\} \text{ for each } y \in \mathcal{N}, \end{aligned}$$

for $x$ we have

$$\begin{aligned} \mathrm{st}^{\rho_b}(x) &= H(x) \cup \mathit{induc}_{\mathcal{P}}\langle D\rangle, \\ \mathrm{bf}^{\rho_b}(x) &= (\mathrm{bf}^{\rho_a}(x) \setminus m) \cup \delta^{i\to x}, \end{aligned}$$

and for each $y \in \mathcal{N} \setminus \{x\}$ we have

$$\begin{aligned} \mathrm{st}^{\rho_b}(y) &= \mathrm{st}^{\rho_a}(y) \\ \mathrm{bf}^{\rho_b}(y) &= \mathrm{bf}^{\rho_a}(y) \cup \delta^{i \to y}. \end{aligned}$$

We say that this transition is *of* the *active* node $x$. Intuitively, the transition expresses how the active node $x$ reads its old state in $\mathrm{st}^{\rho_a}(x)$ together with the received facts in $\mathit{untag}(m)$ (thus without the tags), and then completes this information with subprogram $\mathit{deduc}_{\mathcal{P}}$. Next, the state of $x$ is changed to $\mathrm{st}^{\rho_b}(x)$, which always contains the input facts of $x$, over schema $\mathit{edb}(\mathcal{P})$, and it also includes all facts derived by subprogram $\mathit{induc}_{\mathcal{P}}$, which is applied to the deductive fixpoint. The state $\mathrm{st}^{\rho_b}(x)$ represents that input facts are never lost, and that relations in $\mathit{idb}(\mathcal{P})$ have mutable state, where only the facts that are explicitly derived by $\mathit{induc}_{\mathcal{P}}$ are remembered. Only the state of the active node $x$ changes. Lastly, the subprogram $\mathit{async}_{\mathcal{P}}$ is also applied to the deductive fixpoint. The facts that it generates are called *messages*, and by the syntax of $\mathit{async}_{\mathcal{P}}$, these have an additional location specifier as their first component that indicates the addressee of the message. For each $y \in \mathcal{N}$, we make the set $\delta^{i \to y}$ that contains all messages addressed to $y$: we drop the addressee-component because all facts are destined for $y$, and we attach the send-tag $i$ to the resulting facts. The set $\delta^{i \to y}$ is then added to the buffer of $y$. Messages with an addressee outside the network are ignored. This way of defining local computation closely corresponds to that of the language Webdamlog [1].

A *run* $\mathcal{R}$ *of* $\mathcal{P}$ *on input* $H$ is an infinite sequence of transitions, such that *(i)* the very first configuration is $\mathit{start}(\mathcal{P}, H)$, *(ii)* the output configuration of each transition is the input configuration of the next transition, and *(iii)* the transition at ordinal $i$ of the sequence uses send-tag $i$. The resulting transition system is highly non-deterministic because in each transition we can choose the active node and also what messages from its buffer we want to deliver. An infinite number of transitions is always possible because the set of delivered messages may be empty. If $\mathcal{R}$ is clear from the context, we will typically refer to transitions by their ordinal, and these ordinals start at 0 for technical convenience. Then, for transition $i \in \mathbb{N}$, we write $\rho_i$ to denote its input configuration, $x_i$ to denote the active node and $m_i$ to denote the delivered messages, where $m_i \subseteq \mathrm{bf}^{\rho_i}(x_i)$.

A nice aspect of the operational semantics given here is that every message is just a fact over $\mathit{idb}(\mathcal{P})$. This allows the local Dedalus rules of a recipient node to treat received message facts in the same way as facts in its old state, i.e., there is no noticeable difference. From this viewpoint, communication is in some sense transparent to the nodes, which is one of the design principles of Dedalus.

As a final remark, transitions as they are defined here can simulate *parallel* transitions in which multiple nodes are active at the same time and receive messages from their respective buffers. Indeed, if we would have multiple nodes active during a parallel transition, they would receive messages from their buffers in isolation, and this can be represented by a chain of transitions in which these

nodes receive one after the other precisely the messages that they received in the parallel transition. For this reason, we limit our attention to transitions with single active nodes.

### 3.5 Fairness and Arrival Function

In the literature on process models it is customary to require certain "fairness" conditions on the execution of a system, for instance to exclude some extreme situations that are expected not to happen in reality [14, 10, 23].

Let $\mathcal{P}$ be a Dedalus program. Let $H$ be an input distributed database instance for $\mathcal{P}$, over a network $\mathcal{N}$. A run $\mathcal{R}$ of $\mathcal{P}$ on $H$ is called *fair* if:

- every node is the active node in an infinite number of transitions; and,
- for every transition $i \in \mathbb{N}$, for every $y \in \mathcal{N}$, for every pair $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(y)$, there is a transition $k$ with $i \leq k$ in which $\langle j, \boldsymbol{f} \rangle$ is delivered to $y$, or more formally: $\langle j, \boldsymbol{f} \rangle \in m_k$.

Intuitively, the fairness conditions disallow starvation: every node does an infinite number of local computation steps and every sent message is eventually delivered. We consider only fair runs in this paper.

In the second condition about message deliveries, it is possible that $k = i$, and in that case $\langle j, \boldsymbol{f} \rangle$ is delivered in the transition immediately following configuration $\rho_i$. Because the pair $\langle j, \boldsymbol{f} \rangle$ can be in the message buffer of multiple nodes, this $k$ is not unique for the pair $\langle j, \boldsymbol{f} \rangle$ by itself. But, when we also consider the addressee $y$, it follows from the operational semantics that this $k$ is unique for the triple $(j, y, \boldsymbol{f})$. This reasoning gives rise to a function $\alpha_{\mathcal{R}}$, called the *arrival function for* $\mathcal{R}$, that is defined as follows: for every transition $i$, for every node $y$, for every fact $\langle i, \boldsymbol{f} \rangle \in \delta^{i \to y}$ (i.e., $\boldsymbol{f}$ is sent to addressee $y$ during $i$), the function $\alpha_{\mathcal{R}}$ maps $(i, y, \boldsymbol{f})$ to the transition ordinal $k$ in which $\langle i, \boldsymbol{f} \rangle$ is delivered to $y$. We always have $\alpha_{\mathcal{R}}(i, y, \boldsymbol{f}) > i$. Indeed, the delivery of a message can only happen after it was sent. So, when the delivery of one message causes another to be sent, then the second one is delivered in a later transition. This is related to the topic of "causality", which is discussed in Section 4.3.

## 4 Declarative Semantics

Let $\mathcal{P}$ be a Dedalus program. Throughout this section, we fix $\mathcal{P}$ and give a declarative semantics for this program. In this semantics, we want to abstract away details that are specific to the operational semantics. First, Sections 4.1 and 4.2 will provide additional notations and definitions about runs. These will be used in Section 4.3 to investigate an abstraction of the operational semantics and some of the properties involved. Next, the declarative semantics is given by the stable model semantics applied to a pure Datalog$^\neg$ program $\mathit{pure}(\mathcal{P})$ that is obtained from Dedalus program $\mathcal{P}$. In Sections 4.4 up to 4.8, we describe how to construct $\mathit{pure}(\mathcal{P})$ and we define its semantics. Intuitively, this new program

will simulate the entire distributed computation (of all nodes together) and its construction is centered around the insights obtained in Section 4.3.

## 4.1 Timestamps

We will assign local time values to the steps of a node in the operational semantics. Let $\mathcal{R}$ be a run of $\mathcal{P}$ on some input distributed database instance $H$, which is over a network $\mathcal{N}$. For each transition $i \in \mathbb{N}$ of $\mathcal{R}$, we define $loc_{\mathcal{R}}(i)$ to be the number of transitions in $\mathcal{R}$ with active node $x_i$ that come *strictly* before transition $i$. Note that $loc_{\mathcal{R}}(i)$ is the (zero-based) ordinal of the local step of $x_i$ during transition $i$. We call such step ordinals of a node the *timestamps* of that node, and these can be regarded as local clock values. So, per node, we can identify the local computation steps by their timestamps. We would like to stress that the timestamps are relative to each node. For instance, timestamp 0 for a node $x$ indicates the first step of $x$, and timestamp 0 for another node $y$ indicates the first step of $y$.

As a counterpart to function $loc_{\mathcal{R}}(\cdot)$, for each $(x, s) \in \mathcal{N} \times \mathbb{N}$ we define $glob_{\mathcal{R}}(x, s)$ to be the transition ordinal $i$ of $\mathcal{R}$ such that $x_i = x$ and $loc_{\mathcal{R}}(i) = s$. In words: we find the transition $i$ in which node $x$ does its local computation step with timestamp $s$. It follows from the definition of $loc_{\mathcal{R}}(\cdot)$ that $glob_{\mathcal{R}}(x, s)$ is uniquely defined.

## 4.2 Extended Schema and Trace

We want to associate a *location specifier* and a timestamp to facts over $sch(\mathcal{P})$. Let $R^{(k)} \in sch(\mathcal{P})$. The intuition of a fact $R(x, s, a_1, \ldots, a_k)$ with location specifier $x$ and timestamp $s$ will be that the fact $R(a_1, \ldots, a_k)$ is present at node $x$ during its local step with timestamp $s$, after the program $deduc_{\mathcal{P}}$ is executed (see Section 3.4). For using timestamps in facts, we require that $\mathbb{N} \subseteq \mathbf{dom}$.

Formally, we write $sch(\mathcal{P})^{\mathrm{LT}}$ to denote the database schema obtained from $sch(\mathcal{P})$ by incrementing the arity of every relation by two. The two extra components will contain the location specifier and timestamp, which are by convention the first and second components of a fact.[5]

For a database instance $I$ over $sch(\mathcal{P})$, $x \in \mathbf{dom}$ and $s \in \mathbb{N}$, we write $I^{\Uparrow x,s}$ to denote the facts over $sch(\mathcal{P})^{\mathrm{LT}}$ that are obtained by prepending location specifier $x$ and timestamp $s$ to every fact of $I$. For the reverse operation, for an instance $J$ over $sch(\mathcal{P})^{\mathrm{LT}}$, we write $J^{\Downarrow}$ to denote the facts over $sch(\mathcal{P})$ obtained by removing the location specifier and timestamp from every fact of $J$. Lastly, we write $J|^{x,s}$ to denote the facts of $J$ that have location specifier $x$ and timestamp $s$, without removing the location specifier and timestamp.

When $I$ and $J$ are sets of *atoms* over schemas $sch(\mathcal{P})$ and $sch(\mathcal{P})^{\mathrm{LT}}$ respectively, and $\mathtt{x}, \mathtt{s} \in \mathbf{var}$, we will also apply the notations $I^{\Uparrow \mathtt{x},\mathtt{s}}$ and $J^{\Downarrow}$, with the same meaning as for facts (except that now the location specifiers and timestamps are variables). Also, if $L$ is a sequence of literals over schema $sch(\mathcal{P})$,

[5] The abbreviation "LT" stands for "location specifier and timestamp".

and $\mathtt{x}, \mathtt{s} \in \mathbf{var}$, we write $L^{\Uparrow \mathtt{x},\mathtt{s}}$ to denote the sequence of literals over schema $sch(\mathcal{P})^{\mathrm{LT}}$ that is obtained by adding location specifier $\mathtt{x}$ and timestamp $\mathtt{s}$ to the literals in $L$ (negative literals stay negative).

We will now capture the computed data during a run as a set of facts that we call the *trace*. Let $\mathcal{R}$ be a run of $\mathcal{P}$ on some input $H$, over a network $\mathcal{N}$. For each transition $i \in \mathbb{N}$, let $x_i$ denote the active node, and let $D_i$ denote the output of subprogram $deduc_{\mathcal{P}}$ during $i$. Let $loc_{\mathcal{R}}(i)$ be as defined in Section 4.1. The operational semantics implies that $D_i$ consists of *(i)* the input *edb*-facts at $x_i$; *(ii)* the inductively derived facts during the previous step of $x_i$ (if $loc_{\mathcal{R}}(i) \geq 1$); *(iii)* the messages delivered during transition $i$; and, *(iv)* all facts deductively derived from the previous ones. Intuitively, $D_i$ contains *all* local facts over $sch(\mathcal{P})$ that $x_i$ has during transition $i$. Now, the trace of $\mathcal{R}$ is the following instance over $sch(\mathcal{P})^{\mathrm{LT}}$:

$$trace(\mathcal{R}) = \bigcup_{i \in \mathbb{N}} D_i^{\Uparrow x_i,\, loc_{\mathcal{R}}(i)}.$$

In words: the trace represents all locally computed facts during each transition, additionally carrying the location specifier and timestamp of the active node. The trace shows in detail what happens in the run, in terms of what facts are available on the nodes during which of their steps.

## 4.3 Messages and Causality

In the declarative semantics, we want to represent the same computations as in the operational semantics. We believe that the trace of a run represents in detail the computation of that run (see Section 4.2). So, our goal will be to represent in the declarative semantics exactly the traces of runs. In the operational semantics, we order the actions of the nodes on a fine-grained global time axis, by ordering the transitions in the runs. For representing the trace, we will see below that it is actually sufficient to focus on the direct and indirect relationships between just the local steps of the nodes, ignoring the global ordering of the transitions. This forms the main ingredient for the declarative semantics.

Let $\mathcal{R}$ be a run of $\mathcal{P}$. From $\mathcal{R}$, we extract the message sending and receiving events, or simply called the "message events". Formally, we define $mesg(\mathcal{R})$ to be the set of all tuples $(x, s, y, t, \boldsymbol{f})$, with $\boldsymbol{f} = R(\bar{a})$ a fact, and denoting $i = glob_{\mathcal{R}}(x, s)$ and $j = glob_{\mathcal{R}}(y, t)$, such that $\alpha_{\mathcal{R}}(i, y, \boldsymbol{f}) = j$, i.e., node $x$ during step $s$ sends message $\boldsymbol{f}$ to $y$ that arrives at the step $t$ of $y$, with possibly $x = y$. Note that in $mesg(\mathcal{R})$ there is no mention of transitions since it only contains relationships between local steps. The following lemma says $mesg(\mathcal{R})$ is sufficient for representing the trace:

**Lemma 4.1.** Let $H$ be an input for $\mathcal{P}$. For any two runs $\mathcal{R}_1$ and $\mathcal{R}_2$ of $\mathcal{P}$ on $H$, if $mesg(\mathcal{R}_1) = mesg(\mathcal{R}_2)$ then $trace(\mathcal{R}_1) = trace(\mathcal{R}_2)$.

*Proof.* This result is perhaps not really surprising, and we will only give a proof sketch. Let $x$ be a node of the network that $H$ is over. We will see by inductive

reasoning on the local steps of $x$ that $x$ will produce during each step exactly the same deductive facts in $\mathcal{R}_1$ as in $\mathcal{R}_2$. In the first step of $x$, the previous state of $x$ in both $\mathcal{R}_1$ and $\mathcal{R}_2$ consists of just the local input $H(x)$, since there was no previous step of $x$ to derive inductive facts. Also, the given assumption $\mathit{mesg}(\mathcal{R}_1) = \mathit{mesg}(\mathcal{R}_2)$ implies that $x$ receives exactly the same messages during its first step in $\mathcal{R}_1$ and $\mathcal{R}_2$. Once the previous state and the received messages are known, the execution of the subprograms $\mathit{deduc}_{\mathcal{P}}$ and $\mathit{induc}_{\mathcal{P}}$ during the first step of $x$ is completely determined. Hence, in runs $\mathcal{R}_1$ and $\mathcal{R}_2$ the node $x$ produces exactly the same deductive and inductive facts during the first step. This implies that also the second stored state of $x$ is the same in both runs. Our reasoning can now be repeated for the second step of $x$, the third step, etc. Generalized to all nodes, we see that $\mathit{trace}(\mathcal{R}_1) = \mathit{trace}(\mathcal{R}_2)$. □

Consider now the following example, that illustrates how the pairs in $\mathcal{N} \times \mathbb{N}$ might be related by a run.

**Example 4.2.** Suppose we have a run $\mathcal{R}$, in which the following events take place, where we assume that $x$, $y$ and $z$ are three distinct nodes:

1. Node $x$ during step $s$ sends a message $A$ to node $z$.
2. This message $A$ arrives at node $z$ during step $u$.
3. The node $z$ during step $u + 5$ sends a message $B$ to node $y$.
4. This message $B$ arrives at node $y$ during step $t$.

There is a chain of events that connects $(x, s)$ to $(y, t)$:

$$(x, s) \xrightarrow{\text{send } A} (z, u) \xrightarrow{\text{step}} (z, u + 1) \ldots \xrightarrow{\text{step}} (z, u + 5) \xrightarrow{\text{send } B} (y, t).$$

We say that the step $s$ of node $x$ *(causally) happens before* the step $t$ of node $y$ because we can follow a forward chain of local steps and sent messages to connect step $s$ of $x$ to step $t$ of $y$. And such a path can make a "detour" to other nodes and some of their steps as well; in this case node $z$ and its steps $u$ up to $u + 5$. □

For a run $\mathcal{R}$, the intuition of Example 4.2 can be formalized with the *happens-before* relation [11] on the set $\mathcal{N} \times \mathbb{N}$, which is defined as the smallest relation $\prec_{\mathcal{R}}$ on $\mathcal{N} \times \mathbb{N}$ that satisfies the following three conditions:

- for each $(x, s) \in \mathcal{N} \times \mathbb{N}$, we have $(x, s) \prec_{\mathcal{R}} (x, s + 1)$;
- $(x, s) \prec_{\mathcal{R}} (y, t)$ whenever for some fact $\boldsymbol{f}$ we have $(x, s, y, t, \boldsymbol{f}) \in \mathit{mesg}(\mathcal{R})$;
- $\prec_{\mathcal{R}}$ is transitive, i.e., $(x, s) \prec_{\mathcal{R}} (z, u) \prec_{\mathcal{R}} (y, t)$ implies $(x, s) \prec_{\mathcal{R}} (y, t)$.

We call these three cases respectively *local* edges, *message* edges and *transitive* edges. Naturally, the first two cases express a direct relationship, whereas the third case is more indirect. Note that the relation $\prec_{\mathcal{R}}$ does not say how the

messages are used. For instance, in Example 4.2, we cannot say for sure if $z$ at step $u$ reads the data in message $A$, or that it is "necessary" for $z$ to first receive $A$ before it can send $B$. Also, even if two runs on the same input have the same happens-before relation, it is not guaranteed that they have the same trace. This is because the happens-before relation does not talk about the specific messages that arrive at the nodes (whereas Lemma 4.1 does).

Consider the following property:

**Lemma 4.3.** For every run $\mathcal{R}$, the happens-before relation $\prec_{\mathcal{R}}$ contains no cycles.

*Proof.* If there would be a cycle in $\prec_{\mathcal{R}}$ that contains transitive edges, then we can substitute each transitive edge in this cycle with a path consisting of non-transitive edges. Therefore it is sufficient to show the absence of cycles consisting of only non-transitive edges. We show this with a proof by contradiction. So, suppose that there is a chain in $\mathcal{N} \times \mathbb{N}$ without transitive edges

$$(x_1, s_1) \prec_{\mathcal{R}} (x_2, s_2) \prec_{\mathcal{R}} \ldots \prec_{\mathcal{R}} (x_n, s_n)$$

with $n \geq 2$ and $(x_1, s_1) = (x_n, s_n)$. Because there are no transitive edges, for each $i \in \{1, \ldots, n-1\}$, the edge $(x_i, s_i) \prec_{\mathcal{R}} (x_{i+1}, s_{i+1})$ falls into one of the following two cases:

- $x_i = x_{i+1}$ and $s_{i+1} = s_i + 1$ (local edge);
- $x_i$ during step $s_i$ sends a message to $x_{i+1}$ that arrives in step $s_{i+1}$ of $x_{i+1}$ (message edge).

In the first case, it follows from the definition of $\mathit{loc}_{\mathcal{R}}(\cdot)$ that

$$\mathit{glob}_{\mathcal{R}}(x_i, s_i) < \mathit{glob}_{\mathcal{R}}(x_{i+1}, s_{i+1}).$$

For the second case, by our operational semantics, every message is always delivered in a later transition than the one in which it was sent. So, again we have

$$\mathit{glob}_{\mathcal{R}}(x_i, s_i) < \mathit{glob}_{\mathcal{R}}(x_{i+1}, s_{i+1}).$$

Since this property holds for all the above edges, by transitivity we thus have $\mathit{glob}_{\mathcal{R}}(x_1, s_1) < \mathit{glob}_{\mathcal{R}}(x_n, s_n)$. But that is a contradiction because $(x_1, s_1) = (x_n, s_n)$ and thus $\mathit{glob}_{\mathcal{R}}(x_1, s_1) = \mathit{glob}_{\mathcal{R}}(x_n, s_n)$. □

**Corollary 4.4.** For every run $\mathcal{R}$, the relation $\prec_{\mathcal{R}}$ is a strict partial order on $\mathcal{N} \times \mathbb{N}$.

*Proof.* From its definition, we immediately have that $\prec_{\mathcal{R}}$ is transitive. Secondly, irreflexivity for $\prec_{\mathcal{R}}$ follows from Lemma 4.3. □

In Example 4.2, the happens-before relation is indeed partial because $(x, s)$ does not happen before $(y, t-1)$ and $(y, t-1)$ does not happen before $(x, s)$. So, $(x, s)$ and $(y, t-1)$ are incomparable with $\prec_{\mathcal{R}}$. On a similar note, it is in

general possible that the directed graph with vertices $\mathcal{N} \times \mathbb{N}$ and edges $\prec_{\mathcal{R}}$ is not even weakly connected. This occurs for instance when there are no message edges, in which case the graph consists of only isolated chains of local edges (and their transitive closure).

Consider the following property:

**Corollary 4.5.** For every run $\mathcal{R}$, for each $(x, s, y, t, \boldsymbol{f}) \in \mathit{mesg}(\mathcal{R})$ we have $(y, t) \not\prec_{\mathcal{R}} (x, s)$.

*Proof.* First, $(x, s, y, t, \boldsymbol{f}) \in \mathit{mesg}(\mathcal{R})$ implies $(x, s) \prec_{\mathcal{R}} (y, t)$ by definition of $\prec_{\mathcal{R}}$. So, if $(y, t) \prec_{\mathcal{R}} (x, s)$ then there would be a cycle in $\prec_{\mathcal{R}}$, which not possible by Lemma 4.3. □

This last property is equivalent to saying that if $(y, t) \not\prec_{\mathcal{R}} (x, s)$ then it is possible that $x$ during step $s$ sends a message to $y$ that arrives in step $t$ of $y$. We will concretely use this in our declarative semantics: we only allow $x$ during step $s$ to send a message to $y$ at step $t$ if $(y, t) \not\prec_{\mathcal{R}} (x, s)$ (cf. Section 4.7.3).

## 4.4 Additional Relation Names

In the following subsections, we will start with the construction of $\mathit{pure}(\mathcal{P})$, which is the pure Datalog$^\neg$ program obtained from $\mathcal{P}$. FFor this purpose, we need some new relation names not yet used in $\mathit{sch}(\mathcal{P})$.[6] These are listed in Table 1. The concrete purpose of all these relations will become clear in the following subsections.

| New relation names | Represents |
|---|---|
| `all` | network |
| `time`, `tsucc`, $<$, $\neq$ | timestamps |
| `before` | happens-before relation |
| $\texttt{cand}_R$, $\texttt{chosen}_R$, $\texttt{other}_R$, for each relation name $R$ in $\mathit{idb}(\mathcal{P})$ | messages |
| `hasSender`, `isSmaller`, `hasMax`, `rcvInf` | delivery of only a finite number of messages to each step of a node |

Table 1: Relation names not in $\mathit{sch}(\mathcal{P})$.

## 4.5 Network and Time Relations

In $\mathit{pure}(\mathcal{P})$, we will use unary relation `all` to represent the whole network of interest. For example, a fact $\texttt{all}(x)$ will express that $x$ is a node of the network. A small remark: if the original rules of $\mathcal{P}$ need access to node identifiers, then those identifiers must be explicitly provided in extra input relations different

[6] In practice, this can always be arranged through a namespace mechanism.

from `all` (like relation `Node` in Section 3.1) or they must be received from other nodes by means of messages.

In $\mathit{pure}(\mathcal{P})$, we will also explicitly reason about timestamps, using relations of the following database schema:

$$\mathcal{D}_{\text{time}} = \{\texttt{time}^{(1)}, \texttt{tsucc}^{(2)}, <^{(2)}, \neq^{(2)}\}.$$

The relations '$<$' and '$\neq$' will be written in infix notation in rules. We consider only the following instance over $\mathcal{D}_{\text{time}}$:

$$\begin{aligned} I_{\text{time}} \quad = \quad & \{\texttt{time}(s), \texttt{tsucc}(s, s+1) \mid s \in \mathbb{N}\} \\ & \cup \{(s < t) \mid s, t \in \mathbb{N} : \ s < t\} \\ & \cup \{(s \neq t) \mid s, t \in \mathbb{N} : \ s \neq t\}. \end{aligned}$$

Intuitively, the instance $I_{\text{time}}$ provides timestamps together with relations to compare them.

## 4.6 Representing Causality

We now explain how causality will be represented in $\mathit{pure}(\mathcal{P})$. To express that $(x, s) \in \mathcal{N} \times \mathbb{N}$ causally happens before $(y, t) \in \mathcal{N} \times \mathbb{N}$, we use a fact of the form $\texttt{before}(x, s, y, t)$. We add to $\mathit{pure}(\mathcal{P})$ the following rules:

$$\texttt{before}(\texttt{x}, \texttt{s}, \texttt{x}, \texttt{t}) \leftarrow \texttt{all}(\texttt{x}), \texttt{tsucc}(\texttt{s}, \texttt{t}). \tag{4.1}$$

$$\texttt{before}(\texttt{x}, \texttt{s}, \texttt{y}, \texttt{t}) \leftarrow \texttt{before}(\texttt{x}, \texttt{s}, \texttt{z}, \texttt{u}), \texttt{before}(\texttt{z}, \texttt{u}, \texttt{y}, \texttt{t}). \tag{4.2}$$

The rule (4.1) expresses that on every node, a step causally comes before the next step. The rule (4.2) computes the transitive closure on the `before` relation, because transitivity is required for a partial order.

The above rules are added to $\mathit{pure}(\mathcal{P})$ independently of what rules $\mathcal{P}$ contains. But in the following subsection we will add rules to $\mathit{pure}(\mathcal{P})$ that are obtained by transforming the original rules of $\mathcal{P}$. In particular, the sending of messages will also have an impact on the happens-before relation.

## 4.7 Rule Transformation

For each type of rule in $\mathcal{P}$ we specify what corresponding rules should be added to $\mathit{pure}(\mathcal{P})$. Because $\mathcal{P}$ is constant-free, all rules we add are constant-free. For technical convenience, we will assume that rules of $\mathcal{P}$ always contain at least one positive body atom. This assumption allows us to enforce more elegantly that the variables in the head atoms of $\mathit{pure}(\mathcal{P})$ also occur in at least one positive body atom. This assumption is not really a restriction, since a nullary positive body atom is already sufficient.

First, let $\texttt{x}, \texttt{s}, \texttt{t}, \texttt{t}' \in \mathbf{var}$ be distinct variables that do not yet occur in the rules of $\mathcal{P}$. In $\mathit{pure}(\mathcal{P})$, the variables `x` and `s` will be used as location specifier and timestamp respectively. The variables `t` and $\texttt{t}'$ will also be used for timestamps. We write $\mathbf{B}\{\bar{\texttt{v}}\}$, where $\bar{\texttt{v}}$ is a tuple of variables, to denote any sequence $\beta$ of literals over database schema $\mathit{sch}(\mathcal{P})$, such that the variables in $\beta$ are precisely those in the tuple $\bar{\texttt{v}}$.

### 4.7.1 Deductive Rules

For each deductive rule

$$R(\bar{\mathtt{u}}) \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}\}$$

in $\mathcal{P}$, we add to $\mathit{pure}(\mathcal{P})$ the following rule:

$$R(\mathtt{x}, \mathtt{s}, \bar{\mathtt{u}}) \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}\}^{\Uparrow \mathtt{x},\mathtt{s}}. \tag{4.3}$$

This rule expresses that the facts deductively derived at some node $x$ during step $s$ are (immediately) visible within step $s$ of $x$.

### 4.7.2 Inductive Rules

For each inductive rule

$$R(\bar{\mathtt{u}})\bullet \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}\}$$

in $\mathcal{P}$, we add to $\mathit{pure}(\mathcal{P})$ the following rule:

$$R(\mathtt{x}, \mathtt{t}, \bar{\mathtt{u}}) \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}\}^{\Uparrow \mathtt{x},\mathtt{s}}, \mathtt{tsucc}(\mathtt{s}, \mathtt{t}). \tag{4.4}$$

Intuitively, this rule derives a fact that becomes visible in the *next* step of the *same* node.

### 4.7.3 Asynchronous Rules

For the situation in which a node $x$ at its step $s$ sends a message fact $R(\bar{a})$ to a node $y$, we use a fact $\mathtt{cand}_R(x, s, y, t, \bar{a})$ to say that $t$ could be the arrival timestamp of this message at $y$.[7] We use a fact $\mathtt{chosen}_R(x, s, y, t, \bar{a})$ to say that $t$ is the *effective* arrival timestamp of this message at $y$. Lastly, a fact $\mathtt{other}_R(x, s, y, t, \bar{a})$ means that $t$ is *not* the arrival timestamp of the message.

Now, for each asynchronous rule

$$R(\bar{\mathtt{u}}) \mid \mathtt{y} \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}, \mathtt{y}\}$$

in $\mathcal{P}$, letting $\bar{\mathtt{w}}$ be a tuple of new and distinct variables with $|\bar{\mathtt{w}}| = |\bar{\mathtt{u}}|$, we add to $\mathit{pure}(\mathcal{P})$ the following rules, for which the intuition is given below:

$$\begin{aligned}\mathtt{cand}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{u}}) \leftarrow \mathbf{B}\{\bar{\mathtt{u}}, \bar{\mathtt{v}}, \mathtt{y}\}^{\Uparrow \mathtt{x},\mathtt{s}}, \mathtt{all}(\mathtt{y}), \mathtt{time}(\mathtt{t}),\\ \neg\mathtt{before}(\mathtt{y}, \mathtt{t}, \mathtt{x}, \mathtt{s}).\end{aligned} \tag{4.5}$$

$$\mathtt{chosen}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}) \leftarrow \mathtt{cand}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}), \neg\mathtt{other}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}). \tag{4.6}$$

$$\mathtt{other}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}) \leftarrow \mathtt{cand}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}), \mathtt{chosen}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}', \bar{\mathtt{w}}), \mathtt{t} \neq \mathtt{t}'. \tag{4.7}$$

$$R(\mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}) \leftarrow \mathtt{chosen}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}). \tag{4.8}$$

$$\mathtt{before}(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}) \leftarrow \mathtt{chosen}_R(\mathtt{x}, \mathtt{s}, \mathtt{y}, \mathtt{t}, \bar{\mathtt{w}}). \tag{4.9}$$

[7] Here, '`cand`' abbreviates "candidate".

Rule (4.5) represents the messages that are sent. It evaluates the body of the original asynchronous rule, verifies that the addressee is within the network by using relation `all`, and it generates all possible candidate arrival timestamps that are not restricted by relation `before`. This last restriction comes from Corollary 4.5, and it will prevent cycles from occurring in relation `before`.

Now remains the matter of actually choosing *one* arrival timestamp amongst all these candidates. Intuitively, rule (4.6) selects an arrival timestamp for a message with the condition that this timestamp is not yet ignored, as expressed with relation $\texttt{other}_R$. Also, looking at rule (4.7), a possible arrival timestamp $t$ becomes ignored if there is already a chosen arrival timestamp $t'$ with $t \neq t'$. Together, both rules have the effect that exactly one arrival timestamp will be chosen under the stable model semantics. This technical construction is due to Saccà and Zaniolo [30], who show how to express dynamic choice under the stable model semantics.

Rule (4.8) represents the actual arrival of an $R$-message with the chosen arrival timestamp: the data-tuple in the message becomes part of the addressee's state for relation $R$. When the addressee reads relation $R$, it thus transparently reads the arrived $R$-messages.

The rule (4.9) adds the causal restriction that the local step of the sender happens before the arrival step of the addressee. Together with the previously introduced rules (4.1) and (4.2), this will make sure that when the addressee later *causally* replies to the sender, the reply — as generated by a rule of the form (4.5) — will arrive after this first send-step of the sender.

Note, if multiple asynchronous rules in $\mathcal{P}$ have the same head predicate $R$, only new $\texttt{cand}_R$-rules have to be added because the rules (4.6)–(4.9) are general for all $R$-messages.

Note that if there are asynchronous rules in $\mathcal{P}$, the program $\mathit{pure}(\mathcal{P})$ will not be syntactically stratifiable because relation `before` negatively depends on itself through rules of the following forms, in order: (4.5), (4.6) and (4.9). Moreover, $\mathit{pure}(\mathcal{P})$ is not locally stratifiable [9] because on a network with a least two nodes $x$ and $y$, the fact $\texttt{before}(x, s, y, t)$ with $s, t \in \mathbb{N}$ can negatively depend on itself by means of ground versions of these same rules.

### 4.7.4 Fairness and Finite Messages

We now relate the fairness conditions of Section 3.5 to $\mathit{pure}(\mathcal{P})$. The fairness condition that every node does an infinite number of transitions does not require explicit modeling, because our previous transformations of deductive, inductive and asynchronous rules implicitly look at all possible pairs in $\mathcal{N} \times \mathbb{N}$, i.e., all possible steps for all nodes. The other fairness condition demands that every sent message is eventually delivered. This too is already satisfied by our transformation of the asynchronous rules, because for every sent message we choose precisely one arrival timestamp.

But there is one more thing that requires special attention. Our program $\mathit{pure}(\mathcal{P})$ so far allows an infinite number of messages to arrive at any step of a node. This does not happen in our operational semantics, or in any real-world

distributed system for that matter: indeed, no node has to process an infinite number of messages at any given moment. We consider this to be an additional fairness restriction that must be explicitly enforced in $\mathit{pure}(\mathcal{P})$.

We will approach this problem as follows. Suppose there are an infinite number of messages that arrive at some node $y$ during its step $t$. Since in a network there are only a finite number of nodes and a node can only send a finite number of messages during each step (the active domain is finite), there must be at least one node $x$ that sends messages to step $t$ of $y$ during an infinite number of steps of $x$. Hence there is no maximum value amongst the corresponding send-timestamps of $x$. Thus, in order to prevent the arrival of an infinite number of messages at step $t$ of $y$, it will be sufficient to demand that there always *is* such a maximum send-timestamp for every sender. Below, we will implement this strategy with some concrete rules in $\mathit{pure}(\mathcal{P})$.

We use a fact $\texttt{rcvInf}(y, t)$ to express that node $y$ receives an infinite number of messages during its step $t$. Below we add new rules, and their intuition is that they are relative to an addressee and a step of this addressee, represented by the variables `y` and `t` respectively. First, we add the following rule to $\mathit{pure}(\mathcal{P})$ for *each* relation $\texttt{chosen}_R$ that results from the transformation of asynchronous rules, where `x`, `s`, `y`, and `t` are variables and $\bar{\texttt{w}}$ is a tuple of distinct variables disjoint from the previous ones with $|\bar{\texttt{w}}|$ the arity of relation $R$ in $\mathit{sch}(\mathcal{P})$:

$$\texttt{hasSender}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}) \leftarrow \texttt{chosen}_R(\texttt{x}, \texttt{s}, \texttt{y}, \texttt{t}, \bar{\texttt{w}}), \neg\texttt{rcvInf}(\texttt{y}, \texttt{t}). \tag{4.10}$$

This rule intuitively means that as long as addressee `y` has not received an infinite number of messages during its step `t`, we register the senders and their send-timestamps. Recall that $<^{(2)} \in \mathcal{D}_{\text{time}}$. Next, we add to $\mathit{pure}(\mathcal{P})$ the following rules, for which the intuition is provided below:

$$\begin{aligned}\texttt{isSmaller}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}) \leftarrow \texttt{hasSender}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}), \texttt{hasSender}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}'),\\ \texttt{s} < \texttt{s}'.\end{aligned} \tag{4.11}$$

$$\texttt{hasMax}(\texttt{y}, \texttt{t}, \texttt{x}) \leftarrow \texttt{hasSender}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}), \neg\texttt{isSmaller}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}). \tag{4.12}$$

$$\texttt{rcvInf}(\texttt{y}, \texttt{t}) \leftarrow \texttt{hasSender}(\texttt{y}, \texttt{t}, \texttt{x}, \texttt{s}), \neg\texttt{hasMax}(\texttt{y}, \texttt{t}, \texttt{x}). \tag{4.13}$$

The rule (4.11) checks for each sender and each of its send-timestamps whether there is a later send-timestamp of that same sender. The rule (4.12) tries to find a maximum send-timestamp. Finally, the rule (4.13) derives a `rcvInf`-fact if no maximum send-timestamp was found for at least one sender.

We will show in Section 7.1 that in any stable model, the above rules make sure that every node receives only a finite number of messages at every step.

## 4.8 Input and Stable Models

Now we define the actual declarative semantics for $\mathcal{P}$. Let $H$ be an input distributed database instance for $\mathcal{P}$, over a network $\mathcal{N}$. Let $\mathit{pure}(\mathcal{P})$ be as

previously constructed. We define $decl(H)$ to be the following database instance over the schema $edb(\mathcal{P})^{\text{LT}} \cup \{\texttt{all}^{(1)}\} \cup \mathcal{D}_{\text{time}}$:

$$\begin{aligned} decl(H) \quad = \quad & \{R(x, s, \bar{a}) \mid x \in \mathcal{N},\ s \in \mathbb{N},\ R(\bar{a}) \in H(x)\} \\ & \cup \{\texttt{all}(x) \mid x \in \mathcal{N}\} \cup I_{\text{time}}. \end{aligned}$$

In words: we make for each node its input facts available at all timestamps; we provide the set of all nodes; and, $I_{\text{time}}$ provides the timestamps with comparison relations (see Section 4.5). Note, instance $decl(H)$ is infinite because $\mathbb{N}$ is infinite.

Recall the *stable model semantics* for Datalog$^\neg$ programs, as reviewed in Section 2.2.3. We now define the declarative semantics for $\mathcal{P}$ on input $H$:

**Definition 4.6.** Any stable model of $pure(\mathcal{P})$ on input $decl(H)$ is called a *model* of $\mathcal{P}$ on input $H$.

Importantly, we are using stable models of $pure(\mathcal{P})$, not of $\mathcal{P}$.

# 5 Main Result

Recall from Section 4.2 the definition of the trace of a run, representing in detail the computation of that run. Our main result shows that our declarative semantics of Dedalus expresses exactly the same computations as our operational semantics:

**Theorem 5.1.** Let $\mathcal{P}$ be a Dedalus program and let $H$ be an input distributed database instance for $\mathcal{P}$. On input $H$,

*(i)* for every fair run $\mathcal{R}$ of $\mathcal{P}$ there is a model $M$ of $\mathcal{P}$ such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}$, and

*(ii)* for every model $M$ of $\mathcal{P}$ there is a fair run $\mathcal{R}$ of $\mathcal{P}$ such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}$.

□

The proof of item *(i)* of the theorem is described in Section 6. The proof of item *(ii)*, which is the most difficult, is described in Section 7. We only describe the crucial reasoning steps of the proofs; the intricate technical details can be found in the Appendix.

# 6 Run to Model

Let $\mathcal{P}$ be a Dedalus program and let $H$ be an input distributed database instance for $\mathcal{P}$, over some network $\mathcal{N}$. Let $\mathcal{R}$ be a fair run of $\mathcal{P}$ input $H$. In this section, we show that there is a model $M$ of $\mathcal{P}$ on input $H$ such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}$. The main idea is that we translate the transitions of $\mathcal{R}$ to facts over the schema of $pure(\mathcal{P})$.

## 6.1 Construction

In this section we construct $M$. We define

$$M = \mathit{decl}(H) \cup \bigcup_{i \in \mathbb{N}} \mathrm{trans}_{\mathcal{R}}^{[i]},$$

where $\mathrm{trans}_{\mathcal{R}}^{[i]}$ for each $i \in \mathbb{N}$ is an instance over the schema of $\mathit{pure}(\mathcal{P})$ that describes the transition $i$, which is detailed below. First, the reason for including $\mathit{decl}(H)$ is because if we want $M$ to be a stable model, it must always contain the input $\mathit{decl}(H)$ (see Section 2.2.3). Let $i \in \mathbb{N}$. We define $\mathrm{trans}_{\mathcal{R}}^{[i]}$ as

$$\mathrm{trans}_{\mathcal{R}}^{[i]} = \mathrm{caus}_{\mathcal{R}}^{[i]} \cup \mathrm{fin}_{\mathcal{R}}^{[i]} \cup \mathrm{duc}_{\mathcal{R}}^{[i]} \cup \mathrm{snd}_{\mathcal{R}}^{[i]},$$

where each of these new sets focuses on different aspects of transition $i$, and they are defined next, together with their intuition with respect to $\mathit{pure}(\mathcal{P})$.

Let $\alpha_{\mathcal{R}}$ denote the arrival function of $\mathcal{R}$, as defined in Section 3.5. For the run $\mathcal{R}$, let $\mathit{loc}_{\mathcal{R}}(\cdot)$ and $\mathit{glob}_{\mathcal{R}}(\cdot)$ be as defined in Section 4.1, and let $\prec_{\mathcal{R}}$ be the happens-before relation as defined in Section 4.3. Let us abbreviate $s_i = \mathit{loc}_{\mathcal{R}}(i)$.

**Causality** The set $\mathrm{caus}_{\mathcal{R}}^{[i]}$ represents the pairs $(x, s) \in \mathcal{N} \times \mathbb{N}$ that causally happen before $(x_i, s_i)$. We define $\mathrm{caus}_{\mathcal{R}}^{[i]}$ to consist of all facts $\texttt{before}(x, s, x_i, s_i)$ for which $(x, s) \in \mathcal{N} \times \mathbb{N}$ and $(x, s) \prec_{\mathcal{R}} (x_i, s_i)$. This represents the joint result of rules (4.1), (4.2), and (4.9), corresponding to respectively the local edges, transitive edges, and message edges of $\prec_{\mathcal{R}}$.

**Finite Messages** The set $\mathrm{fin}_{\mathcal{R}}^{[i]}$ represents that only a finite number of messages are delivered in transition $i$, thus at step $s_i$ of node $x_i$. First, let $\mathrm{senders}_{\mathcal{R}}^{[i]}$ denote all pairs $(x, s) \in \mathcal{N} \times \mathbb{N}$ such that, denoting $j = \mathit{glob}_{\mathcal{R}}(x, s)$, for some fact $\boldsymbol{f}$ we have $\alpha_{\mathcal{R}}(j, x_i, \boldsymbol{f}) = i$, i.e., the node $x$ during its step $s$ sends a message to $x_i$ with arrival timestamp $s_i$. It follows from the operational semantics that for each $(x, s) \in \mathrm{senders}_{\mathcal{R}}^{[i]}$ we have $\mathit{glob}_{\mathcal{R}}(x, s) < i$.

We define $\mathrm{fin}_{\mathcal{R}}^{[i]}$ to consist of the following facts:

- the fact $\texttt{hasSender}(x_i, s_i, x, s)$ for each $(x, s) \in \mathrm{senders}_{\mathcal{R}}^{[i]}$, representing the result of rule (4.10);
- the fact $\texttt{isSmaller}(x_i, s_i, x, s)$ for each $(x, s) \in \mathrm{senders}_{\mathcal{R}}^{[i]}$ and $(x, s') \in \mathrm{senders}_{\mathcal{R}}^{[i]}$ with $s < s'$, representing the result of rule (4.11);
- the fact $\texttt{hasMax}(x_i, s_i, x)$ for each sender-node $x$ mentioned in $\mathrm{senders}_{\mathcal{R}}^{[i]}$, representing the result of rule (4.12).

We know that in $\mathcal{R}$ only a finite number of messages arrive at step $s_i$ of $x_i$. Hence we add no fact $\texttt{rcvInf}(x_i, s_i)$ to $\mathrm{fin}_{\mathcal{R}}^{[i]}$. This also explains why the specification of the `hasMax`-facts above is relatively simple: there is always a maximum send-timestamp for each sender-node.

**Deductive** The set $\text{duc}_{\mathcal{R}}^{[i]}$ represents all facts over $sch(\mathcal{P})$ that are available at $x_i$ during transition $i$, thus during step $s_i$ of $x_i$. Let $D_i$ denote the output of subprogram $deduc_{\mathcal{P}}$ during transition $i$. We define $\text{duc}_{\mathcal{R}}^{[i]}$ to consist of the facts $D_i^{\Uparrow x_i, s_i}$. This represents the result of rules in $pure(\mathcal{P})$ of the form (4.3), (4.4) and (4.8).

**Sending** The set $\text{snd}_{\mathcal{R}}^{[i]}$ represents the sending of messages during transition $i$. Let $\text{mesg}_{\mathcal{R}}^{[i]}$ denote the output of subprogram $async_{\mathcal{P}}$ during transition $i$, restricted to the facts having their addressee-component in the network (see Section 3.3).

We define $\text{snd}_{\mathcal{R}}^{[i]}$ to consist of the following facts:

- the fact $\texttt{cand}_R(x_i, s_i, y, t, \bar{a})$ for each $R(y, \bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$ and $t \in \mathbb{N}$ such that $(y, t) \not\prec_{\mathcal{R}} (x_i, s_i)$, representing the result of rule (4.5);
- all facts $\texttt{chosen}_R(x_i, s_i, y, t, \bar{a})$ and $\texttt{other}_R(x_i, s_i, y, u, \bar{a})$ for which $R(y, \bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$, $t = loc_{\mathcal{R}}(j)$ with $j = \alpha_{\mathcal{R}}(i,\ y,\ R(\bar{a}))$, $u \in \mathbb{N}$, $(y, u) \not\prec_{\mathcal{R}} (x_i, s_i)$ and $u \neq t$. This represents the choice of an arrival timestamp for the messages, as performed by rules (4.6) and (4.7).

## 6.2 Conclusion

In Appendix A it is shown that $M$ is a model of $\mathcal{P}$ on input $H$. By construction of $M$, we have, as desired:

$$M|_{sch(\mathcal{P})^{\text{LT}}} = \bigcup_{i \in \mathbb{N}} \text{duc}_{\mathcal{R}}^{[i]} = \bigcup_{i \in \mathbb{N}} D_i^{\Uparrow x_i, s_i} = trace(\mathcal{R}).$$

# 7 Model to Run

Let $\mathcal{P}$ be a Dedalus program and let $H$ be an input distributed database instance for $\mathcal{P}$, over some network $\mathcal{N}$. Let $M$ be a model of $\mathcal{P}$ on input $H$. In this section we show that there is a fair run $\mathcal{R}$ of $\mathcal{P}$ on input $H$ such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}$.

The direction shown in Section 6 is perhaps the most intuitive direction because we only have to show that a concrete set of facts is actually a stable model. In this section we do not yet understand what $M$ can contain. So, a first important step is to show that $M$ has some desirable properties which allow us to construct a run from it.

First, it is important to know that in $M$ we find location specifiers where we expect location specifiers and we find timestamps where we expect timestamps. Formally, we call $M$ *well-formed* if:

- for each $R(x, s, \bar{a}) \in M|_{sch(\mathcal{P})^{\text{LT}}}$ we have $x \in \mathcal{N}$ and $s \in \mathbb{N}$;
- for each $\texttt{before}(x, s, y, t) \in M$, we have $x, y \in \mathcal{N}$ and $s, t \in \mathbb{N}$;

- for each fact $\texttt{cand}_R(x, s, y, t, \bar{a})$, $\texttt{chosen}_R(x, s, y, t, \bar{a})$ and $\texttt{other}_R(x, s, y, t, \bar{a})$ in $M$, we have $x, y \in \mathcal{N}$ and $s, t \in \mathbb{N}$;
- for each fact $\texttt{hasSender}(x, s, y, t)$, $\texttt{isSmaller}(x, s, y, t)$, $\texttt{hasMax}(x, s, y)$ and $\texttt{rcvInf}(x, s)$ in $M$, we have $x, y \in \mathcal{N}$ and $s, t \in \mathbb{N}$.

Using the notation from Section 2.2.3, let $gr_M^{\mathcal{P},H}$ abbreviate the ground program

$$ground_M(pure(\mathcal{P}), decl(H)).$$

By definition of $M$ as a stable model, we have $M = gr_M^{\mathcal{P},H}(decl(H))$. It can be shown by induction on the fixpoint computation of $gr_M^{\mathcal{P},H}$ on input $decl(H)$ that $M$ is always well-formed. We omit the details.

## 7.1 Partial Order

Based on the relation $\texttt{before}$ in $M$, in this subsection we define a strict partial order $\prec_M$ on $\mathcal{N} \times \mathbb{N}$. This forms a crucial insight in the causality information represented by $M$. In the following subsection, we use this partial order to establish a total order on $\mathcal{N} \times \mathbb{N}$, around which we can build a run. The idea is that this total order tells us which are the active nodes in the transitions of the constructed run.

We define $\prec_M$ as follows. For each $(x, s) \in \mathcal{N} \times \mathbb{N}$ and $(y, t) \in \mathcal{N} \times \mathbb{N}$, we write $(x, s) \prec_M (y, t)$ if and only if $\texttt{before}(x, s, y, t) \in M$.

Regarding terminology, an edge $(x, s) \prec_M (y, t)$ is called a *local edge*, a *message edge* or a *transitive edge* if the fact $\texttt{before}(x, s, y, t) \in M$ can be derived by a ground rule in $gr_M^{\mathcal{P},H}$ of respectively the form (4.1), the form (4.9), or the form (4.2).[8] It is possible that an edge is of two or even three types at the same time.

The rest of this section is dedicated to showing that $\prec_M$ has certain desirable properties, so that we can later derive a total order with desirable properties. First, consider the following claim:

**Claim 7.1.** Relation $\prec_M$ is a strict partial order on $\mathcal{N} \times \mathbb{N}$.

*Proof.* We show that $\prec_M$ is transitive and irreflexive.

**Transitive** First, we show that $\prec_M$ is transitive. Suppose we have $(x, s) \prec_M (z, u)$ and $(z, u) \prec_M (y, t)$. We have to show that $(x, s) \prec_M (y, t)$. We have $\texttt{before}(x, s, z, u) \in M$ and $\texttt{before}(z, u, y, t) \in M$. Because the rule (4.2) is positive, we have the following ground rule in $gr_M^{\mathcal{P},H}$:

$$\texttt{before}(x, s, y, t) \leftarrow \texttt{before}(x, s, z, u), \texttt{before}(z, u, y, t).$$

Because $M$ is a stable model and the body of the previous ground rule is in $M$, we obtain $\texttt{before}(x, s, y, t) \in M$. Hence, $(x, s) \prec_M (y, t)$, as desired.

[8] The body of such a ground rule has to be in $M$.

**Irreflexive** Because an edge $(x, s) \prec_M (x, s)$ for any $(x, s) \in \mathcal{N} \times \mathbb{N}$ would form a cycle of length one, it is sufficient to show that there are no cycles in $\prec_M$ at all. This gives us irreflexivity, as desired.

First, let $\prec'_M$ denote the restriction of $\prec_M$ to the edges that are local or message edges. Note that this definition allows some edges in $\prec'_M$ to also be transitive. The edges that are missing from $\prec'_M$ with respect to $\prec_M$ are only derivable by ground rules of the form (4.2); we call these the *pure* transitive edges. We start by showing that $\prec'_M$ contains no cycles. We show this with a proof by contradiction. So, suppose that there is a cycle in $\mathcal{N} \times \mathbb{N}$ through the edges of $\prec'_M$:

$$(x_1, s_1) \prec_M (x_2, s_2) \prec_M \ldots \prec_M (x_n, s_n)$$

with $n \geq 2$ and $(x_1, s_1) = (x_n, s_n)$. We have $\texttt{before}(x_i, s_i, x_{i+1}, s_{i+1}) \in M$ for each $i \in \{1, \ldots, n-1\}$. Moreover, based on these previous `before`-facts, ground rules in $gr_M^{\mathcal{P},H}$ of the form (4.2) will have derived $\texttt{before}(x_i, s_i, x_j, s_j) \in M$ for each $i, j \in \{1, \ldots, n\}$. If each edge on the above cycle would be only local, then for each $i, j \in \{1, \ldots, n\}$ with $i < j$ we have $x_i = x_j$ and $s_i < s_j$, and hence $s_1 \neq s_n$, which is false. So, there has to be some $k \in \{1, \ldots, n-1\}$ such that $(x_k, s_k) \prec_M (x_{k+1}, s_{k+1})$ is a message edge, derived by a ground rule of the form (4.9):

$$\texttt{before}(x_k, s_k, x_{k+1}, s_{k+1}) \leftarrow \texttt{chosen}_R(x_k, s_k, x_{k+1}, s_{k+1}, \bar{a}).$$

Therefore $\texttt{chosen}_R(x_k, s_k, x_{k+1}, s_{k+1}, \bar{a}) \in M$. This $\texttt{chosen}_R$-fact must be derived by a ground rule of the form (4.6) in $gr_M^{\mathcal{P},H}$, which implies that

$$\texttt{cand}_R(x_k, s_k, x_{k+1}, s_{k+1}, \bar{a}) \in M.$$

This $\texttt{cand}_R$-fact must in turn be derived by a ground rule $\psi$ of the form (4.5). Because rules of the form (4.5) in $\mathit{pure}(\mathcal{P})$ contain a negative `before`-atom in their body, the presence of $\psi$ in $gr_M^{\mathcal{P},H}$ requires that $\texttt{before}(x_{k+1}, s_{k+1}, x_k, s_k) \notin M$. But that is a contradiction, because $\texttt{before}(x_i, s_i, x_j, s_j) \in M$ for each $i, j \in \{1, \ldots, n\}$ (see above).

Now we show there are no cycles in the entire relation $\prec_M$. Since $M = gr_M^{\mathcal{P},H}(\mathit{decl}(H))$, we have $M = \bigcup_{i \in \mathbb{N}} M_i$ where $M_0 = \mathit{decl}(H)$ and $M_i = T(M_{i-1})$ for each $i \geq 1$ where $T$ is the immediate consequence operator of $gr_M^{\mathcal{P},H}$. By induction on $i$, we show that an edge $\texttt{before}(x, s, y, t) \in M_i$ either is a local or message edge, or it can be replaced by a path of local or message edges in $M_i$. Then any cycle in $\prec_M$ would imply there is a cycle in $\prec'_M$, which is impossible. So, $\prec_M$ can not contain cycles. Now, this induction property is satisfied for the base case because $M_0$ does not contain `before`-facts. For the induction hypothesis, assume the property holds for $M_{i-1}$, where $i \geq 1$. For the inductive step, let $\texttt{before}(x, s, y, t) \in M_i \setminus M_{i-1}$. If this fact is derived by a ground rule of the form (4.1) or (4.9) then the property is satisfied. Now suppose the fact is derived by a ground rule of the form (4.2):

$$\texttt{before}(x, s, y, t) \leftarrow \texttt{before}(x, s, z, u), \texttt{before}(z, u, y, t).$$

Both body facts are in $M_{i-1}$, implying $M_{i-1}$ contains a path of local or message edges from $(x, s)$ to $(z, u)$ and from $(z, u)$ to $(y, t)$. Hence, using $M_{i-1} \subseteq M_i$, the edge $\texttt{before}(x, s, y, t) \in M_i$ can be replaced by a path of local or message edges in $M_i$. □

In Section 4.7.4 we have added extra rules to $pure(\mathcal{P})$ to enforce that every node only receives a finite number of messages during each step. We now verify that this works correctly:

**Claim 7.2.** For each $(y, t) \in \mathcal{N} \times \mathbb{N}$ there are only a finite number of pairs $(x, s) \in \mathcal{N} \times \mathbb{N}$ such that $(x, s) \prec_M (y, t)$ is a message edge.

*Proof.* We start by noting that $M$ does not contain the fact $\texttt{rcvInf}(y, t)$. Indeed, in order to derive this fact, we need a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.13), which has a body fact of the form $\texttt{hasSender}(y, t, x, s)$. Such $\texttt{hasSender}$-facts must be generated by ground rules in $gr_M^{\mathcal{P},H}$ of the form (4.10). The rule (4.10) negatively depends on relation $\texttt{rcvInf}$. Thus, specifically, if we want a ground rule in $gr_M^{\mathcal{P},H}$ that can derive $\texttt{hasSender}(y, t, x, s)$, we should require the absence of $\texttt{rcvInf}(y, t)$ from $M$. So $\texttt{rcvInf}(y, t) \in M$ requires $\texttt{rcvInf}(y, t) \notin M$, which is impossible.

The rest of the proof works towards a contradiction. So, suppose that $(y, t)$ has an infinite number of incoming message edges. Because there are only a finite number of nodes in $\mathcal{N}$, there has to be a node $x$ that has an infinite number of timestamps $s$ such that $\texttt{before}(x, s, y, t) \in M$ is a message edge. Since it is a message edge, such a fact $\texttt{before}(x, s, y, t)$ can be generated by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.9), which implies that there is a relation $R$ in $idb(\mathcal{P})$ and a tuple $\bar{a}$ such that $\texttt{chosen}_R(x, s, y, t, \bar{a}) \in M$. Because $\texttt{rcvInf}(y, t) \notin M$ (see above), for *each* of these $\texttt{chosen}_R$-facts, there is a ground rule of the form (4.10) in $M$ that derives $\texttt{hasSender}(y, t, x, s) \in M$.

Rule (4.13) has a negative $\texttt{hasMax}$-atom in its body. If we can show that $\texttt{hasMax}(y, t, x) \notin M$, then there will be a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.13), where $\texttt{hasSender}(y, t, x, s) \in M$:

$$\texttt{rcvInf}(y, t) \leftarrow \texttt{hasSender}(y, t, x, s).$$

This then causes $\texttt{rcvInf}(y, t) \in M$, giving the desired contradiction.

Also towards a proof by contradiction, suppose that $\texttt{hasMax}(y, t, x) \in M$. This means that there is a ground rule $\psi$ in $gr_M^{\mathcal{P},H}$ of the form (4.12):

$$\texttt{hasMax}(y, t, x) \leftarrow \texttt{hasSender}(y, t, x, s).$$

Because the rule (4.12) contains a negative $\texttt{isSmaller}$-atom in the body, and because $\psi \in gr_M^{\mathcal{P},H}$, we know that $\texttt{isSmaller}(y, t, x, s) \notin M$. But because there are infinitely many facts of the form $\texttt{hasSender}(y, t, x, s') \in M$, there is at least one fact $\texttt{hasSender}(y, t, x, s') \in M$ with $s < s'$. Moreover, the rule (4.11) is positive, and therefore the following ground rule is always in $gr_M^{\mathcal{P},H}$:

$$\texttt{isSmaller}(y, t, x, s) \leftarrow \texttt{hasSender}(y, t, x, s), \texttt{hasSender}(y, t, x, s'),\ s < s'.$$

Since the body of this ground rule is in $M$, the rule derives $\texttt{isSmaller}(y, t, x, s) \in M$, which gives the desired contradiction. □

An ordering $\prec$ on a set $A$ is called *well-founded* if for each $a \in A$, there are only a finite number of elements $b \in A$ such that $b \prec a$. We now use Claim 7.2 to show:

**Claim 7.3.** Relation $\prec_M$ on $\mathcal{N} \times \mathbb{N}$ is well-founded.

*Proof.* Let $(x, s) \in \mathcal{N} \times \mathbb{N}$. We have to show that there are only a finite number of pairs $(y, t) \in \mathcal{N} \times \mathbb{N}$ such that $(y, t) \prec_M (x, s)$. Technically, we can limit our attention to paths in $\prec_M$ consisting of local edges and message edges, because if we can show that there are only a finite number of predecessors of $(x, s)$ on such paths, then there are only a finite number of predecessors when we include the transitive edges as well. First we show that every pair $(y, t) \in \mathcal{N} \times \mathbb{N}$ has only a finite number of incoming local and message edges. If $t > 0$, we can immediately see that $(y, t)$ has precisely one incoming local edge, as created by a ground rule of the form (4.1), and if $t = 0$ then $(y, t)$ has no incoming local edge. Also, Claim 7.2 tells us that $(y, t)$ has only a finite number of incoming message edges. So, the number of incoming local and message edges in $(y, t)$ is finite.

Let $(y, t) \in \mathcal{N} \times \mathbb{N}$ be a pair such that $(y, t) \prec_M (x, s)$ is a local edge or a message edge. Starting in $(x, s)$, we can follow this edge backwards so that we reach $(y, t)$. If $(y, t)$ itself has incoming local or message edges, from $(y, t)$ we can again follow an edge backwards. This way we can incrementally construct backward paths starting from $(x, s)$. Because at each pair of $\mathcal{N} \times \mathbb{N}$ there are only a finite number of incoming local or message edges (shown above), if $(x, s)$ would have an infinite number of predecessors, we must be able to construct a backward path of infinite length. We now show that the existence of such an infinite path leads to a contradiction. So, suppose that there is a backward path of infinite length. Because there are only a finite number of nodes in the network $\mathcal{N}$, there must be a node $y$ that occurs infinitely often on this path. We will now show that, as we progress further along the backward path, we must see the local timestamps of $y$ strictly decrease. Hence, we must eventually reach timestamp 0 of $y$, after which we cannot decrement the timestamps of $y$ anymore, and thus it is impossible that $y$ occurs infinitely often along the path. Suppose that the timestamps of $y$ do not strictly decrease. There are two cases. First, if the same pair $(y, t)$ would occur twice on the path, we would have a cycle in $\prec_M$, which is not possible by Claim 7.1. Secondly, suppose that there are two timestamps $t$ and $t'$ of $y$ such that $t < t'$ and $(y, t)$ occurs before $(y, t')$ on the backward path, meaning that $(y, t)$ lies closer to $(x, s)$. Because the edges were followed in reverse, we have

$$(y, t') \prec_M \ldots \prec_M (y, t).$$

But since $t < t'$, by means of local edges, we always have

$$(y, t) \prec_M (y, t + 1) \prec_M \ldots \prec_M (y, t').$$

Combining these two sets of edges leads to a cycle, which is impossible by Claim 7.1. □

## 7.2 Construction of Run

Let $\prec_M$ be the well-founded strict partial order on $\mathcal{N} \times \mathbb{N}$ as defined in the preceding subsection. The relation $\prec_M$ has the intuition of a happens-before relation of a run (Section 4.3), but the novelty is that it comes from a purely declarative model $M$. We will now use $\prec_M$ to construct a run $\mathcal{R}$ such that $\mathit{trace}(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}$.

It is well-known that a well-founded strict partial order can be extended to a well-founded strict total order. So, let $<_M$ be a well-founded strict total order on $\mathcal{N} \times \mathbb{N}$ that extends $\prec_M$, i.e., for each $(x, s) \in \mathcal{N} \times \mathbb{N}$ and $(y, t) \in \mathcal{N} \times \mathbb{N}$, if $(x, s) \prec_M (y, t)$ then $(x, s) <_M (y, t)$, but the reverse does not have to hold.

Ordering the set $\mathcal{N} \times \mathbb{N}$ according to $<_M$ gives us a sequence of pairs that will form the transitions in the constructed run $\mathcal{R}$. Concretely, we obtain a sequence of nodes by taking the node-component from each pair. This will form our sequence of active nodes. Similarly, by taking the timestamp-component from each pair of $\mathcal{N} \times \mathbb{N}$, we obtain a sequence of timestamps. These are the local clocks of the active nodes during their transitions.

We introduce some extra notations to help us reason about the ordering of time that is implied by $<_M$. For each $(x, s) \in \mathcal{N} \times \mathbb{N}$, let $\mathit{glob}_M(x, s) \in \mathbb{N}$ denote the ordinal of $(x, s)$ as implied by $<_M$, which is well-defined because $<_M$ is well-founded. For technical convenience, we let ordinals start at 0. Note that $\mathit{glob}_M(\cdot)$ is an injective function. For any $i \in \mathbb{N}$, we define $(x_i, s_i)$ to be the unique pair in $\mathcal{N} \times \mathbb{N}$ such that $\mathit{glob}_M(x_i, s_i) = i$.

As a counterpart to function $\mathit{glob}_M(\cdot)$, for $i \in \mathbb{N}$ and $x \in \mathcal{N}$, let $\mathit{loc}_M(i, x)$ denote the *size* of the set

$$\{(x, s) \in \mathcal{N} \times \mathbb{N} \mid \mathit{glob}_M(x, s) < i\}.$$

Intuitively, if $i$ is regarded to be the ordinal of a transition in a run, the number $\mathit{loc}_M(i, x)$ is the number of local steps of $x$ that came before transition $i$, i.e., the number of transitions before $i$ in which $x$ was the active node. If $x = x_i$ (the active node) then $\mathit{loc}_M(i, x)$ is effectively the timestamp of $x$ *during* transition $i$, and if $x \neq x_i$ then $\mathit{loc}_M(i, x)$ is the next timestamp of $x$ that still has to come after transition $i$. Note that the functions $\mathit{glob}_M(\cdot)$ and $\mathit{loc}_{\mathcal{R}}(\cdot)$ closely resemble the functions $\mathit{glob}_{\mathcal{R}}(\cdot)$ and $\mathit{loc}_{\mathcal{R}}(\cdot)$ of Section 4.1.

We will now define the desired run $\mathcal{R}$ of $\mathcal{P}$ on $H$. First we define the (infinite) sequence of configurations $\rho_0$, $\rho_1$, $\rho_2$, etc. In a second step we will connect each pair of subsequent configurations by a transition. Recall from Section 3.2 that a configuration describes for each node what facts it has stored locally (state), and also what messages have been sent to this node but that are not yet received (message buffer). The facts that are stored on a node are either input *edb*-facts, or facts derived by inductive rules in a previous step of the node. The first kind of facts can be easily obtained from $M$ by keeping only the facts over schema

$edb(\mathcal{P})^{\mathrm{LT}}$, which gives a subset of $decl(H)$. For the second kind of facts, we look at the inductively derived facts in $M$, which is detailed next. The rules in $pure(\mathcal{P})$ that represent inductive rules of $\mathcal{P}$ are easily recognizable: they are of the form (4.4), meaning that they have a head atom over $sch(\mathcal{P})^{\mathrm{LT}}$ and they have a positive body `tsucc`-atom. No other kind of rule in $pure(\mathcal{P})$ has this form. Hence, the ground rules in $gr_M^{\mathcal{P},H}$ that are based on rules of the form (4.4) are also easily recognizable, and we will call these *inductive ground rules*. A ground rule $\psi \in gr_M^{\mathcal{P},H}$ is called *active* on $M$ if $pos_\psi \subseteq M$, which implies that $head_\psi \in M$ because $M$ is stable. Let $M^{\mathrm{ind}}$ denote all head atoms of inductive ground rules in $gr_M^{\mathcal{P},H}$ that are active on $M$. Note that $M^{\mathrm{ind}} \subseteq M$. Now, for each $i \in \mathbb{N}$, for each node $x \in \mathcal{N}$, denoting $s = loc_M(i,x)$, in configuration $\rho_i$ the state $\mathrm{st}^{\rho_i}(x)$ is defined as

$$\left((M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x,s} \cup M^{\mathrm{ind}}|^{x,s}\right)^{\Downarrow}.$$

Using notations from Section 4.2, note that we remove the location specifier and timestamp because we have to obtain facts over the schema of $\mathcal{P}$, not over the schema of $pure(\mathcal{P})$.

Now we define the message buffers in the configurations. Recall that the message buffer of a node always contains pairs of the form $\langle j, \boldsymbol{f}\rangle$, where $j \in \mathbb{N}$ is the transition in which the fact $\boldsymbol{f}$ over $idb(\mathcal{P})$ was sent. For each $i \in \mathbb{N}$, for each node $x \in \mathcal{N}$, in configuration $\rho_i$ the message buffer $\mathrm{bf}^{\rho_i}(x)$ is defined as

$$\{\langle glob_M(y,t),\, R(\bar{a})\rangle \mid \\ \exists u : \mathtt{chosen}_R(y,t,x,u,\bar{a}) \in M,\, glob_M(y,t) < i \leq glob_M(x,u)\}.$$

Note the use of addressee $x$ in the definition of $\mathrm{bf}^{\rho_i}(x)$. The definition of $\mathrm{bf}^{\rho_i}(x)$ reflects the operational semantics, in that the messages in the buffer of node $x$ must be sent in a previous transition, as expressed by the constraint $glob_M(y,t) < i$. Moreover, the constraint $i \leq glob_M(x,u)$ says that $\mathrm{bf}^{\rho_i}(x)$ contains only messages that will be delivered in transitions of $x$ that come after configuration $\rho_i$. Possibly $i = glob_M(x,u)$, and in that case the message will be delivered in the transition immediately after configuration $\rho_i$ (see also below).

So far we have obtained a sequence of configurations $\rho_0$, $\rho_1$, $\rho_2$, etc. Now we define a sequence of tuples, one tuple per ordinal $i \in \mathbb{N}$, that represents the transition $i$. Let $i \in \mathbb{N}$. The tuple $\tau_i$ is defined as $(\rho_i, x_i, m_i, i, \rho_{i+1})$, also denoted as $\rho_i \xrightarrow[i]{x_i,\, m_i} \rho_{i+1}$, where

$$m_i = \{\langle glob_M(y,t),\, R(\bar{a})\rangle \mid \mathtt{chosen}_R(y,t,z,u,\bar{a}) \in M,\, glob_M(z,u) = i\}.$$

Intuitively, in $m_i$, we select all messages that arrive in transition $i$. And since $glob_M(z,u) = i$ implies $z = x_i$ and $u = s_i$, we thus select all messages destined for step $s_i$ of node $x_i$.

In Appendix B it is shown that the sequence $\mathcal{R}$ is indeed a legal run of $\mathcal{P}$ on input $H$ such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\mathrm{LT}}}$. In the following subsection we show that $\mathcal{R}$ is also fair.

### 7.3 Fair Run

In this subsection we show that run $\mathcal{R}$ is fair. Recall from Section 3.5 that we have to check two fairness conditions:

1. every node is the active node in an infinite number of transitions; and,
2. for every transition $i \in \mathbb{N}$, for every node $y \in \mathcal{N}$, for every $\langle j, \boldsymbol{f} \rangle \in \text{bf}^{\rho_i}(y)$, there is a transition $k$ with $i \leq k$ in which $\langle j, \boldsymbol{f} \rangle$ is delivered (to $y$).

We show that the first fairness condition is satisfied by $\mathcal{R}$. Let $x \in \mathcal{N}$ be a node, and let $s \in \mathbb{N}$ be a timestamp of $x$. Consider transition $i = \mathit{glob}_M(x, s)$. This transition has active node $x_i = x$. We can find such a transition with active node $x$ for every timestamp $s \in \mathbb{N}$ of $x$, and these transitions are all unique because function $\mathit{glob}_M(\cdot)$ is injective. So, there are an infinite number of transitions in $\mathcal{R}$ with active node $x$, and the first fairness condition is satisfied.

Now we show that the second fairness condition is satisfied. Let $i \in \mathbb{N}$, $y \in \mathcal{N}$, and $\langle j, \boldsymbol{f} \rangle \in \text{bf}^{\rho_i}(y)$. Denote $\boldsymbol{f} = R(\bar{a})$. By definition of $\text{bf}^{\rho_i}(y)$, $\langle j, \boldsymbol{f} \rangle \in \text{bf}^{\rho_i}(y)$ implies that there are values $x \in \mathcal{N}$, $s \in \mathbb{N}$ and $t \in \mathbb{N}$ such that $\texttt{chosen}_R(x, s, y, t, \bar{a}) \in M$ and $j = \mathit{glob}_M(x, s) < i \leq \mathit{glob}_M(y, t)$. Denote $k = \mathit{glob}_M(y, t)$. Hence, $i \leq k$ and $\langle j, \boldsymbol{f} \rangle \in m_k$ by definition of $m_k$. Thus $\langle j, \boldsymbol{f} \rangle$ is delivered to $x_k = y$ in transition $k$, as desired.

## 8 Conclusion and Future Work

In the literature, many operational semantics for declarative networking languages were previously developed [13, 28, 17, 7, 1]. In the current work, we have focused on the language Dedalus [5, 6, 19], and we have shown that distributed computations expressed in an operational way can also be described purely declaratively, based on stable models.

Regarding future work, in this paper we have probably not yet explored the full power of stable models. We therefore expect that this work can be extended to languages that incorporate more powerful language constructs, such as dynamic choice [22], aggregation [26], or constructs that allow for reasoning about different time-scales on which events occur [16]. It might also be possible to remove the syntactic stratification condition that we used for the deductive rules of Dedalus.

More related to multi-agent systems [25, 29, 24], it might be interesting to allow logic programs used in declarative networking to dynamically modify their rules. The question would be how (and if) this can be represented in a declarative semantics.

Lastly, we can think about the output of Dedalus programs. Marczak et al. [27] define the output of Dedalus programs with *ultimate* facts, which are facts that will eventually always be present on the network. This way, the output of a run (or equivalently stable model) can be defined. Then, a *consistent* Dedalus program is required to produce the same output in every run. For consistent programs, the output on an input distributed database instance can

thus be defined as the output of any run. We can now consider the following decision problem: for a consistent Dedalus program, an input distributed database instance for that program, and a fact, decide if this fact is output by the program on that input. We think that decidability depends on the semantics of the message buffers. In this paper, we represented per addressee duplicate messages in its message buffer. This is a realistic representation, since in a real network, the same message can be sent multiple times, and hence, multiple instances of the same message can be in transmission simultaneously. If in the message buffers we would remove duplicate messages, then the decision problem becomes decidable because only a finite number of configurations would be possible by finiteness of the input domain. But when duplicates are preserved, the number of configurations is not limited, and we expect that the problem will be undecidable in general. However, we might want to investigate whether decidability can be obtained in particular (syntactically defined) cases. If so, it might be interesting for those cases to find finite representations of the stable models. This could serve as a more intuitive programmer abstraction, or it could perhaps be used to more efficiently simulate the behavior of the network for testing purposes.

## Acknowledgment

The last author thanks Serge Abiteboul for a number of interesting discussions.

## References

[1] S. Abiteboul, M. Bienvenu, A. Galland, et al. A rule-based language for Web data management. In *Proceedings 30th ACM Symposium on Principles of Database Systems*, pages 293–304. ACM Press, 2011.

[2] S. Abiteboul, R. Hull, and V. Vianu. *Foundations of Databases*. Addison-Wesley, 1995.

[3] J.J. Alferes, L.M. Pereira, H. Przymusinska, and T.C. Przymusinski. LUPS—a language for updating logic programs. *Artificial Intelligence*, 138(1–2):87–116, 2002.

[4] P. Alvaro, N. Conway, J. Hellerstein, and W.R. Marczak. Consistency analysis in Bloom: A CALM and collected approach. In *Proceedings 5th Biennial Conference on Innovative Data Systems Research*, pages 249–260. www.cidrdb.org, 2011.

[5] P. Alvaro, W. Marczak, et al. Dedalus: Datalog in time and space. Technical Report EECS-2009-173, University of California, Berkeley, 2009.

[6] P. Alvaro, W.R. Marczak, et al. Dedalus: Datalog in time and space. In de Moor et al. [12], pages 262–281.

[7] T.J. Ameloot, F. Neven, and J. Van den Bussche. Relational transducers for declarative networking. In *Proceedings 30th ACM Symposium on Principles of Database Systems*, pages 283–292. ACM Press, 2011.

[8] T.J. Ameloot and J. Van den Bussche. Deciding eventual consistency for a simple class of relational transducers. In *Proceedings of the 15th International Conference on Database Theory*, pages 86–98. ACM Press, 2012.

[9] K.R. Apt and R.N. Bol. Logic programming and negation: A survey. *The Journal of Logic Programming*, 19-20, Supplement 1(0):9–71, 1994.

[10] K.R. Apt, N. Francez, and S. Katz. Appraising fairness in languages for distributed programming. *Distributed Computing*, 2:226–241, 1988.

[11] H. Attiya and J. Welch. *Distributed Computing: Fundamentals, Simulations, and Advanced Topics*. Wiley, 2004.

[12] O. de Moor, G. Gottlob, T. Furche, and A. Sellers, editors. *Datalog Reloaded: First International Workshop, Datalog 2010*, volume 6702 of *Lecture Notes in Computer Science*, 2011.

[13] A. Deutsch, L. Sui, V. Vianu, and D. Zhou. Verification of communicating data-driven Web services. In *Proceedings 25th ACM Symposium on Principles of Database Systems*, pages 90–99. ACM Press, 2006.

[14] N. Francez. *Fairness*. Springer-Verlag New York, Inc., New York, NY, USA, 1986.

[15] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In *Proceedings of the Fifth International Conference on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[16] G. Greco, A. Guzzo, D. Saccà, and F. Scarcello. Event choice datalog: a logic programming language for reasoning in multiple dimensions. In *Proceedings of the 6th ACM SIGPLAN International Conference on Principles and Practice of Declarative Programming*, PPDP, pages 238–249. ACM Press, 2004.

[17] S. Grumbach and F. Wang. Netlog, a rule-based language for distributed programming. In M. Carro and R. Peña, editors, *Proceedings 12th International Symposium on Practical Aspects of Declarative Languages*, volume 5937 of *Lecture Notes in Computer Science*, pages 88–103, 2010.

[18] J.M. Hellerstein. Datalog redux: experience and conjecture. Video available (under the title "The Declarative Imperative") from `http://db.cs.berkeley.edu/jmh/`, 2010. PODS 2010 keynote.

[19] J.M. Hellerstein. The declarative imperative: experiences and conjectures in distributed logic. *SIGMOD Record*, 39(1):5–19, 2010.

[20] S.S. Huang, T.J. Green, and B.T. Loo. Datalog and emerging applications: an interactive tutorial. In *Proceedings of the 2011 ACM SIGMOD International Conference on the Management of Data*, SIGMOD '11, pages 1213–1216. ACM, 2011.

[21] Trevor Jim. Sd3: A trust management system with certified evaluation. *Security and Privacy, IEEE Symposium on*, 0:106–115, 2001.

[22] R Krishnamurthy and S.A. Naqvi. Non-deterministic choice in datalog. In *Proceedings of the Third International Conference on Data and Knowledge Bases*, pages 416–424, 1988.

[23] L. Lamport. Fairness and hyperfairness. *Distributed Computing*, 13:239–245, November 2000.

[24] J. Leite and L. Soares. Adding evolving abilities to a multi-agent system. In *Proceedings of the 7th International Conference on Computational Logic in Multi-agent Systems*, CLIMA VII'06, pages 246–265. Springer-Verlag, 2007.

[25] J.A. Leite, J.J. Alferes, and L.M. Pereira. Minerva - a dynamic logic programming agent architecture. In *Revised Papers from the 8th International Workshop on Intelligent Agents VIII*, ATAL, pages 141–157. Springer-Verlag, 2002.

[26] B.T. Loo et al. Declarative networking. *Communications of the ACM*, 52(11):87–95, 2009.

[27] W. Marczak, P. Alvaro, N. Conway, J.M. Hellerstein, and D. Maier. Confluence analysis for distributed programs: A model-theoretic approach. Technical Report UCB/EECS-2011-154, EECS Department, University of California, Berkeley, Dec 2011.

[28] J.A. Navarro and A. Rybalchenko. Operational semantics for declarative networking. In A. Gill and T. Swift, editors, *Proceedings 11th International Symposium on Practical Aspects of Declarative Languages*, volume 5419 of *Lecture Notes in Computer Science*, pages 76–90, 2009.

[29] V. Nigam and J. Leite. A dynamic logic programming based system for agents with declarative goals. In *Proceedings of the 4th International Conference on Declarative Agent Languages and Technologies*, DALT, pages 174–190. Springer-Verlag, 2006.

[30] D. Saccà and C. Zaniolo. Stable models and non-determinism in logic programs with negation. In *Proceedings of the Ninth ACM Symposium on Principles of Database Systems*, pages 205–217. ACM Press, 1990.

[31] M. Vardi. The complexity of relational query languages. In *Proceedings 14th ACM Symposium on the Theory of Computing*, pages 137–146, 1982.

[32] D. Zinn, T.J. Green, and B. Ludaescher. Win-move is coordination-free. In *Proceedings of the 15th International Conference on Database Theory*, pages 99–113. ACM Press, 2012.

# Appendix

## General Remarks

Let $\mathcal{P}$ be a Dedalus program. Recall from Section 3.3 that $\mathit{deduc}_\mathcal{P} \subseteq \mathcal{P}$ is the subset of all (unmodified) deductive rules. The semantics of $\mathit{deduc}_\mathcal{P}$ is given by the stratified semantics. Although the semantics of $\mathit{deduc}_\mathcal{P}$ does not depend on the chosen syntactic stratification, for technical convenience in the proofs, we will fix an arbitrary syntactic stratification for $\mathit{deduc}_\mathcal{P}$. Whenever we refer to the stratum number of a relation in $\mathit{idb}(\mathit{deduc}_\mathcal{P})$, it is assumed that this fixed syntactic stratification is used to determine the stratum number. Recall that stratum numbers start at 1.

# A Run to Model: Proof Details

This section contains the proof details of Section 6. Recall the symbols defined there. In this section we show that the constructed set $M$ is a model of $\mathcal{P}$ on input $H$.

Also recall the definition of stable model semantics from Section 2.2.3. Let $gr_M^{\mathcal{P},H}$ abbreviate the ground program

$$\mathit{ground}_M(\mathit{pure}(\mathcal{P}), \mathit{decl}(H)).$$

This program can be considered even if $M$ is not stable. Denote $N = gr_M^{\mathcal{P},H}(\mathit{decl}(H))$. We must show that $N = M$. The inclusions $M \subseteq N$ and $N \subseteq M$ are shown respectively in Sections A.1 and A.2.

We will use the following notations for run $\mathcal{R}$. For each $i \in \mathbb{N}$, recall that transition $i$ transforms configuration $\rho_i$ into configuration $\rho_{i+1}$. Let $x_i$ and $m_i \subseteq \mathrm{bf}^{\rho_i}(x_i)$ denote respectively the active node and the received messages during transition $i$. Also, let $s_i$ abbreviate $\mathit{loc}_\mathcal{R}(i)$, and let $D_i$ denote the output of subprogram $\mathit{deduc}_\mathcal{P}$ during transition $i$.

### A.1 First Inclusion

In this section we show that $M \subseteq N$. By definition,

$$M = \mathit{decl}(H) \cup \bigcup_{i \in \mathbb{N}} \mathrm{trans}_\mathcal{R}^{[i]}.$$

First, we immediately have $\mathit{decl}(H) \subseteq N$ by the semantics of $gr_M^{\mathcal{P},H}$.

Next, we define for uniformity the set $\mathrm{trans}_\mathcal{R}^{[-1]} = \emptyset$. We will show by induction on $i = -1, 0, 1, \ldots$, that $\mathrm{trans}_\mathcal{R}^{[i]} \subseteq N$. The base case ($i = -1$) is trivial. For the induction hypothesis, let $i \geq 0$, and assume for all $j \in \{-1, 0, \ldots, i-1\}$ that $\mathrm{trans}_\mathcal{R}^{[j]} \subseteq N$. We show that $\mathrm{trans}_\mathcal{R}^{[i]} \subseteq N$. By definition,

$$\mathrm{trans}_\mathcal{R}^{[i]} = \mathrm{caus}_\mathcal{R}^{[i]} \cup \mathrm{fin}_\mathcal{R}^{[i]} \cup \mathrm{duc}_\mathcal{R}^{[i]} \cup \mathrm{snd}_\mathcal{R}^{[i]}.$$

We will show inclusion in $N$ for each of these four sets. The numbered claims we will refer to can be found in Section A.1.1.

**Causality** We show that $\text{caus}_{\mathcal{R}}^{[i]} \subseteq N$. Let $(x, s) \in \mathcal{N} \times \mathbb{N}$ be a pair such that $(x, s) \prec_{\mathcal{R}} (x_i, s_i)$. We have to show that $\texttt{before}(x, s, x_i, s_i) \in N$. We distinguish between the following cases:

- Suppose that $(x, s)$ happens before $(x_i, s_i)$ by means of a local edge, i.e., $x = x_i$ and $s_i = s + 1$.

  Because the rule (4.1) in $\mathit{pure}(\mathcal{P})$ is positive, the following ground rule is always in $gr_M^{\mathcal{P},H}$:

  $$\texttt{before}(x, s, x, s + 1) \leftarrow \texttt{all}(x), \texttt{tsucc}(s, s + 1).$$

  This rule derives $\texttt{before}(x, s, x, s + 1) = \texttt{before}(x, s, x_i, s_i) \in N$ because its body facts are in $\mathit{decl}(H) \subseteq N$.

- Suppose that $(x, s)$ happens before $(x_i, s_i)$ by means of a message edge, i.e., there is an earlier transition $j < i$ with $j = \mathit{glob}_{\mathcal{R}}(x, s)$, in which $x$ sends a message $\boldsymbol{f}$ to $x_i$ such that $\alpha_{\mathcal{R}}(j, x_i, \boldsymbol{f}) = i$. Denote $\boldsymbol{f} = R(\bar{a})$.

  Because rules of the form (4.9) in $\mathit{pure}(\mathcal{P})$ are positive, the following ground rule is always in $gr_M^{\mathcal{P},H}$:

  $$\texttt{before}(x, s, x_i, s_i) \leftarrow \texttt{chosen}_R(x, s, x_i, s_i, \bar{a}).$$

  We now show that the body of this rule is in $N$, so that $\texttt{before}(x, s, x_i, s_i) \in N$, as desired. Using $x_j = x$, $s_j = s$, and $s_i = \mathit{loc}_{\mathcal{R}}(i)$, we have

  $$\texttt{chosen}_R(x, s, x_i, s_i, \bar{a}) \in \text{snd}_{\mathcal{R}}^{[j]} \subseteq \text{trans}_{\mathcal{R}}^{[j]}.$$

  By applying the induction hypothesis to transition $j$, we have $\text{trans}_{\mathcal{R}}^{[j]} \subseteq N$, and obtain:

  $$\texttt{chosen}_R(x, s, x_i, s_i, \bar{a}) \in N.$$

- Suppose that $(x, s)$ happens before $(x_i, s_i)$ not by a local edge or message edge, but by a path containing minimally two edges. Let $(z, u)$ denote the penultimate pair on the path so that $(x, s) \prec_{\mathcal{R}} (z, u)$ (by transitivity) and $(z, u) \prec_{\mathcal{R}} (x_i, s_i)$, where the latter relation happens via a local edge or a message edge.

  Because the rule (4.2) is positive, the following ground rule is always in $gr_M^{\mathcal{P},H}$:

  $$\texttt{before}(x, s, x_i, s_i) \leftarrow \texttt{before}(x, s, z, u), \texttt{before}(z, u, x_i, s_i).$$

  We will now show that the body of this rule is in $N$, which implies $\texttt{before}(x, s, x_i, s_i) \in N$, as desired. First, denote $j = \mathit{glob}_{\mathcal{R}}(z, u)$. Because $(z, u) \prec_{\mathcal{R}} (x_i, s_i)$, using the operational semantics, it can be shown

that $j < i$. And because $(x, s) \prec_{\mathcal{R}} (z, u)$, we have $\texttt{before}(x, s, z, u) \in \text{caus}_{\mathcal{R}}^{[j]}$. By applying the induction hypothesis to $j$, we therefore have $\texttt{before}(x, s, z, u) \in N$. Secondly, because $(z, u) \prec_{\mathcal{R}} (x_i, s_i)$ is a local edge or a message edge, we have $\texttt{before}(z, u, x_i, s_i) \in N$ (shown in the preceding two cases).

**Finite Messages** We show that $\text{fin}_{\mathcal{R}}^{[i]} \subseteq N$. Let $\text{senders}_{\mathcal{R}}^{[i]}$ be as defined in Section 6. For each kind of fact in $\text{fin}_{\mathcal{R}}^{[i]}$, we show inclusion in $N$.

Let $\texttt{hasSender}(x_i, s_i, x, s) \in \text{fin}_{\mathcal{R}}^{[i]}$. We have $(x, s) \in \text{senders}_{\mathcal{R}}^{[i]}$, which means that $x$ during step $s$ sends a message fact $R(\bar{a})$ that arrives in step $s_i$ of $x_i$. Rules in $\mathit{pure}(\mathcal{P})$ of the form (4.10) have a negative $\texttt{rcvInf}$-atom in their body. But because $\texttt{rcvInf}(x_i, s_i)$ is not added to $\text{trans}_{\mathcal{R}}^{[i]}$, and hence not to $M$, the following rule is in $gr_M^{\mathcal{P},H}$:

$$\texttt{hasSender}(x_i, s_i, x, s) \leftarrow \texttt{chosen}_R(x, s, x_i, s_i, \bar{a}).$$

We show that the single body fact is in $N$, so the rule derives $\texttt{hasSender}(x_i, s_i, x, s) \in N$, as desired. Denote $j = \mathit{glob}_{\mathcal{R}}(x, s)$. Using that $x = x_j$ and $s = s_j$, we have $\texttt{chosen}_R(x, s, x_i, s_i, \bar{a}) \in \text{snd}_{\mathcal{R}}^{[j]}$. Because $j < i$ by the operational semantics, we can apply the induction hypothesis to $j$ to know that $\text{snd}_{\mathcal{R}}^{[j]} \subseteq N$, and thus $\texttt{chosen}_R(x, s, x_i, s_i, \bar{a}) \in N$, as desired.

Let $\texttt{isSmaller}(x_i, s_i, x, s) \in \text{fin}_{\mathcal{R}}^{[i]}$. We have $(x, s) \in \text{senders}_{\mathcal{R}}^{[i]}$ and there is a timestamp $s' \in \mathbb{N}$ so that $(x, s') \in \text{senders}_{\mathcal{R}}^{[i]}$ and $s < s'$. The rule (4.11) is positive and therefore the following ground rule is always in $gr_M^{\mathcal{P},H}$:

$$\begin{aligned}\texttt{isSmaller}(x_i, s_i, x, s) \quad \leftarrow \quad & \texttt{hasSender}(x_i, s_i, x, s), \texttt{hasSender}(x_i, s_i, x, s'), \\ & s < s'.\end{aligned}$$

We immediately have $(s < s') \in \mathit{decl}(H) \subseteq N$. By construction of $\text{fin}_{\mathcal{R}}^{[i]}$, we have $\texttt{hasSender}(x_i, s_i, x, s) \in \text{fin}_{\mathcal{R}}^{[i]}$ and $\texttt{hasSender}(x_i, s_i, x, s') \in \text{fin}_{\mathcal{R}}^{[i]}$, and thus both facts are also in $N$ as was shown above. Hence the previous ground rule derives $\texttt{isSmaller}(x_i, s_i, x, s) \in N$.

Let $\texttt{hasMax}(x_i, s_i, x) \in \text{fin}_{\mathcal{R}}^{[i]}$. Thus $x$ is a sender-node mentioned in $\text{senders}_{\mathcal{R}}^{[i]}$. Let $s$ be the maximum send-timestamp of $x$ in $\text{senders}_{\mathcal{R}}^{[i]}$, which surely exists because $\text{senders}_{\mathcal{R}}^{[i]}$ is finite. There is no timestamp $s'$ such that $(x, s') \in \text{senders}_{\mathcal{R}}^{[i]}$ and $s < s'$. Therefore we have not added the fact $\texttt{isSmaller}(x_i, s_i, x, s)$ to $\text{fin}_{\mathcal{R}}^{[i]}$, and thus also not to $M$. Although the rule (4.12) contains a negated $\texttt{isSmaller}$-atom, $\texttt{isSmaller}(x_i, s_i, x, s) \notin M$ implies that the following ground is in $gr_M^{\mathcal{P},H}$:

$$\texttt{hasMax}(x_i, s_i, x) \leftarrow \texttt{hasSender}(x_i, s_i, x, s).$$

Moreover, $(x, s) \in \text{senders}_{\mathcal{R}}^{[i]}$ implies $\texttt{hasSender}(x_i, s_i, x, s) \in N$, and thus the previous ground rule derives $\texttt{hasMax}(x_i, s_i, x) \in N$, as desired.

**Deductive** We show that $\mathrm{duc}_{\mathcal{R}}^{[i]} \subseteq N$. By definition, $\mathrm{duc}_{\mathcal{R}}^{[i]} = D_i^{\Uparrow x_i, s_i}$. We will show that $D_i^{\Uparrow x_i, s_i} \subseteq N$. Recall from Section 3.4 that the input of subprogram $\mathit{deduc}_{\mathcal{P}}$ during transition $i$ consists of the set

$$\mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i).$$

If we can show that $(\mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i))^{\Uparrow x_i, s_i} \subseteq N$, then we can apply Claim A.1 to know that $D_i^{\Uparrow x_i, s_i} \subseteq N$, as desired.

We first show that $\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i, s_i} \subseteq N$. There are two cases to consider:

- Suppose that $s_i = 0$, i.e., $i$ is the first transition with active node $x_i$. Then there were no previous state modifications of $x_i$, and thus $\mathrm{st}^{\rho_i}(x_i) = H(x_i)$, which are just the input facts for $x_i$. In that case $\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i, s_i} \subseteq N$ because $H(x_i)^{\Uparrow x_i, s_i} \subseteq \mathit{decl}(H) \subseteq N$.
- Suppose that $s_i > 0$. This means that $x_i$ has done some previous transitions. Let $j$ be the last transition of $x_i$ that came before $i$. Since there are no other transitions of $x_i$ between $j$ and $i$, it follows from the operational semantics that $\mathrm{st}^{\rho_i}(x_i) = \mathrm{st}^{\rho_{j+1}}(x_j)$. By applying the induction hypothesis to $j$, we have $\mathrm{duc}_{\mathcal{R}}^{[j]} \subseteq \mathrm{trans}_{\mathcal{R}}^{[j]} \subseteq N$. Using that $x_j = x_i$ and $s_j + 1 = s_i$, we can apply Claim A.3 to know that $\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i, s_i} = \mathrm{st}^{\rho_{j+1}}(x_j)^{\Uparrow x_j, s_j + 1} \subseteq N$.

Now we show that $\mathit{untag}(m_i)^{\Uparrow x_i, s_i} \subseteq N$. Let $\boldsymbol{f} \in \mathit{untag}(m_i)$. We have to show that $\boldsymbol{f}^{\Uparrow x_i, s_i} \in N$. First, because $\boldsymbol{f} \in \mathit{untag}(m_i)$, there is a transition $k$ with $k < i$ such that $\langle k, \boldsymbol{f} \rangle \in m_i$, i.e., the fact $\boldsymbol{f}$ was sent to $x_i$ during transition $k$ (by node $x_k$). Denote $\boldsymbol{f} = R(\bar{a})$. So, there must be an asynchronous rule with head-predicate $R$ in $\mathcal{P}$, which has a corresponding rule in $\mathit{pure}(\mathcal{P})$ of the form (4.8). Rules of the form (4.8) are positive and thus the following ground rule is always in $gr_M^{\mathcal{P},H}$:

$$R(x_i, s_i, \bar{a}) \leftarrow \mathtt{chosen}_R(x_k, s_k, x_i, s_i, \bar{a}).$$

We show that the body of this rule is in $N$, so the rule derives $\boldsymbol{f}^{\Uparrow x_i, s_i} \in N$, as desired. Because $x_k$ sends $\boldsymbol{f}$ to $x_i$ during transition $k$, and $i$ is the transition in which this message is delivered to $x_i$, we have $\mathtt{chosen}_R(x_k, s_k, x_i, s_i, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[k]} \subseteq \mathrm{trans}_{\mathcal{R}}^{[k]}$. By applying the induction hypothesis to $k$, we have $\mathrm{snd}_{\mathcal{R}}^{[k]} \subseteq N$ and thus $\mathtt{chosen}_R(x_k, s_k, x_i, s_i, \bar{a}) \in N$.

**Sending** We show that $\mathrm{snd}_{\mathcal{R}}^{[i]} \subseteq N$. For each kind of fact in $\mathrm{snd}_{\mathcal{R}}^{[i]}$ we show inclusion in $N$.

Let $\mathtt{cand}_R(x_i, s_i, y, t, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]}$. We have $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$, $t \in \mathbb{N}$ and $(y, t) \not\sim_{\mathcal{R}} (x_i, s_i)$. Since $D_i^{\Uparrow x_i, s_i} \subseteq N$ (see above), we can use Claim A.4 to obtain $\mathtt{cand}_R(x_i, s_i, y, t, \bar{a}) \in N$, as desired.

Let $\mathtt{chosen}_R(x_i, s_i, y, t, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]}$. We have $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and $t = \mathit{loc}_{\mathcal{R}}(j)$ with $j = \alpha_{\mathcal{R}}(i, y, R(\bar{a}))$. Because $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$, this fact was produced by $\mathit{async}_{\mathcal{P}}$, and thus there is an asynchronous rule in $\mathcal{P}$ with head-predicate $R$. This asynchronous rule has a corresponding rule in $\mathit{pure}(\mathcal{P})$ of the

form (4.6), that contains a negated $\texttt{other}_R$-atom in the body. But by construction of $\text{snd}^{[i]}_{\mathcal{R}}$, we have not added $\texttt{other}_R(x_i, s_i, y, t, \bar{a})$ to $\text{snd}^{[i]}_{\mathcal{R}}$, and thus also not to $M$. Therefore the following ground rule of the form (4.6) is in $gr^{\mathcal{P},H}_M$:

$$\texttt{chosen}_R(x_i, s_i, y, t, \bar{a}) \leftarrow \texttt{cand}_R(x_i, s_i, y, t, \bar{a}).$$

Because $j > i$ by the operational semantics, we have $(y, t) \not\prec_{\mathcal{R}} (x_i, s_i)$. Thus, by construction of $\text{snd}^{[i]}_{\mathcal{R}}$, we have $\texttt{cand}_R(x_i, s_i, y, t, \bar{a}) \in \text{snd}^{[i]}_{\mathcal{R}}$, in which case $\texttt{cand}_R(x_i, s_i, y, t, \bar{a}) \in N$ (shown above). Hence, the previous ground rule derives $\texttt{chosen}_R(x_i, s_i, y, t, \bar{a}) \in N$, as desired.

Let $R(y, \bar{a})$ and $t$ be from above. Let $\texttt{other}_R(x_i, s_i, y, u, \bar{a}) \in \text{snd}^{[i]}_{\mathcal{R}}$. We have $u \in \mathbb{N}$, $(y, u) \not\prec_{\mathcal{R}} (x_i, s_i)$ and $u \neq t$. Because the rule(4.7) is positive, the following ground rule is in $gr^{\mathcal{P},H}_M$:

$$\begin{aligned}\texttt{other}_R(x_i, s_i, y, u, \bar{a}) \quad \leftarrow \quad & \texttt{cand}_R(x_i, s_i, y, u, \bar{a}),\ \texttt{chosen}_R(x_i, s_i, y, t, \bar{a}),\\ & u \neq t.\end{aligned}$$

We immediately have $(u \neq t) \in \mathit{decl}(H) \subseteq N$. Now we show that the other body facts are in $N$, so the rule derives $\texttt{other}_R(x_i, s_i, y, u, \bar{a}) \in N$, as desired. Because $(y, u) \not\prec_{\mathcal{R}} (x_i, s_i)$, by construction of $\text{snd}^{[i]}_{\mathcal{R}}$, we have $\texttt{cand}_R(x_i, s_i, y, u, \bar{a}) \in \text{snd}^{[i]}_{\mathcal{R}}$ and thus $\texttt{cand}_R(x_i, s_i, y, u, \bar{a}) \in N$ (shown above). Moreover, it was shown above that $\texttt{chosen}_R(x_i, s_i, y, t, \bar{a}) \in N$.

### A.1.1 Subclaims

**Claim A.1.** Let $i$ be a transition of $\mathcal{R}$. If $(\text{st}^{\rho_i}(x) \cup \mathit{untag}(m_i))^{\Uparrow x_i, s_i} \subseteq N$, then $D_i^{\Uparrow x_i, s_i} \subseteq N$.

*Proof.* Abbreviate $I_i = \text{st}^{\rho_i}(x) \cup \mathit{untag}(m_i)$. Recall that $D_i = \mathit{deduc}_{\mathcal{P}}(I_i)$, which is computed with the stratified semantics.

For a number $k$, we write $D_i^{\to k}$ to denote the set obtained by adding to $I_i$ all facts derived in stratum 1 up to stratum $k$ during the computation of $D_i$. For the largest stratum number $n$ of $\mathit{deduc}_{\mathcal{P}}$, we have $D_i^{\to n} = D_i$. Also, because stratum numbers start at 1, we have $D_i^{\to 0} = I_i$. We show by induction on $k = 0$, 1, 2, …, $n$, that $(D_i^{\to k})^{\Uparrow x_i, s_i} \subseteq N$.

**Base case** For the base case, $k = 0$, the property holds by the given assumption $I_i^{\Uparrow x_i, s_i} \subseteq N$.

**Induction hypothesis** For the induction hypothesis, assume for some stratum number $k$ with $k \geq 1$ that $(D_i^{\to k-1})^{\Uparrow x_i, s_i} \subseteq N$.

**Inductive step** For the inductive step, we show that $(D_i^{\to k})^{\Uparrow x_i, s_i} \subseteq N$. Recall that the input of stratum $k$ in $\mathit{deduc}_{\mathcal{P}}$ is the set $D_i^{\to k-1}$, and the semantics is given by the fixpoint semantics of semi-positive Datalog$^{\neg}$ (see Section 2.2.2). So, we can consider $D_i^{\to k}$ to be a fixpoint, i.e., as the set $\bigcup_{l \in \mathbb{N}} A_l$ with $A_0 =$

$D_i^{\to k-1}$ and $A_l = T(A_{l-1})$ for each $l \geq 1$, where $T$ is the immediate consequence operator of stratum $k$. We show by inner induction on $l = 0$, 1, etc, that

$$(A_l)^{\Uparrow x_i, s_i} \subseteq N.$$

For the base case ($l = 0$), we have $A_0 = D_i^{\to k-1}$, for which we can apply the outer induction hypothesis to know that $(D_i^{\to k-1})^{\Uparrow x_i, s_i} = (A_0)^{\Uparrow x_i, s_i} \subseteq N$, as desired. For the inner induction hypothesis, we assume for some $l \geq 1$ that $(A_{l-1})^{\Uparrow x_i, s_i} \subseteq N$. For the inner inductive step, we show that $(A_l)^{\Uparrow x_i, s_i} \subseteq N$. Let $\boldsymbol{f} \in A_l \setminus A_{l-1}$. Let $\varphi \in \mathit{deduc}_{\mathcal{P}}$ and $V$ be a rule from stratum $k$ and valuation respectively that have derived $\boldsymbol{f}$. Let $\varphi'$ be the rule in $\mathit{pure}(\mathcal{P})$ obtained by applying the transformation (4.3) to $\varphi$. Let $V'$ be $V$ extended to assign $x_i$ and $s_i$ to the new variables in $\varphi'$ that represent the location and timestamp respectively. Note in particular that $V'(\mathit{pos}_{\varphi'}) = V(\mathit{pos}_{\varphi})^{\Uparrow x_i, s_i}$ and $V'(\mathit{neg}_{\varphi'}) = V(\mathit{neg}_{\varphi})^{\Uparrow x_i, s_i}$. Let $\psi$ be the *positive* ground rule obtained by applying $V'$ to $\varphi'$ and by subsequently removing all negative (ground) body atoms. We show that $\psi \in \mathit{gr}_M^{\mathcal{P},H}$ and that its body is in $N$, so that $\psi$ derives $\mathit{head}_\psi = \boldsymbol{f}^{\Uparrow x_i, s_i} \in N$, as desired.

- In order for $\psi$ to be in $\mathit{gr}_M^{\mathcal{P},H}$, it is required that $V'(\mathit{neg}_{\varphi'}) \cap M = \emptyset$. Because $V$ is satisfying for $\varphi$, and negation in $\varphi$ is only applied to lower strata, we have $V(\mathit{neg}_{\varphi}) \cap D_i^{\to k-1} = \emptyset$. Moreover, since a relation is computed in only one stratum of $\mathit{deduc}_{\mathcal{P}}$, we overall have $V(\mathit{neg}_{\varphi}) \cap D_i = \emptyset$. Then by Claim A.2 we have $V(\mathit{neg}_{\varphi})^{\Uparrow x_i, s_i} \cap M = \emptyset$. Hence,

  $$V'(\mathit{neg}_{\varphi'}) \cap M = \emptyset.$$

- Now we show that $\mathit{pos}_\psi \subseteq N$. Because $V$ is satisfying for $\varphi$, we have $V(\mathit{pos}_{\varphi}) \subseteq A_{l-1}$, and by applying the inner induction hypothesis we have $V(\mathit{pos}_{\varphi})^{\Uparrow x_i, s_i} \subseteq N$. Therefore, $\mathit{pos}_\psi = V'(\mathit{pos}_{\varphi'}) \subseteq N$.

□

---

**Claim A.2.** Let $i$ be a transition. Let $I$ be a set of facts over $\mathit{sch}(\mathcal{P})$. If $I \cap D_i = \emptyset$ then $I^{\Uparrow x_i, s_i} \cap M = \emptyset$.

*Proof.* For every fact $\boldsymbol{f} \in M$ that is over schema $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$, and has location specifier $x_i$ and timestamp $s_i$, we have $\boldsymbol{f} \in \mathrm{duc}_{\mathcal{R}}^{[i]}$ because *(i)* for any transition $j$ there are no facts over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$ in $\mathrm{caus}_{\mathcal{R}}^{[j]}$, $\mathrm{fin}_{\mathcal{R}}^{[j]}$ or $\mathrm{snd}_{\mathcal{R}}^{[j]}$; *(ii)* we only add facts with location specifier $x_i$ to $\mathrm{duc}_{\mathcal{R}}^{[j]}$ if $j$ is a transition of node $x_i$; and, *(iii)* for every transition $j$ of node $x_i$, the local timestamp $s_j = \mathit{loc}_{\mathcal{R}}(j)$ is different from $s_i$ whenever $i \neq j$.

Hence, it suffices to show $I^{\Uparrow x_i, s_i} \cap \mathrm{duc}_{\mathcal{R}}^{[i]} = \emptyset$. But this is immediate from $I \cap D_i = \emptyset$ because $\mathrm{duc}_{\mathcal{R}}^{[i]}$ equals $D_i^{\Uparrow x_i, s_i}$ by definition. □

**Claim A.3.** Let $j$ be a transition, and suppose that $\mathrm{duc}_{\mathcal{R}}^{[j]} \subseteq N$. Recall that transition $j$ transforms configuration $\rho_j$ into configuration $\rho_{j+1}$. We have $\mathrm{st}^{\rho_{j+1}}(x_j)^{\Uparrow x_j, s_j+1} \subseteq N$.

*Proof.* By the operational semantics, we have

$$\mathrm{st}^{\rho_{j+1}}(x_j) = H(x_j) \cup \mathit{induc}_{\mathcal{P}}\langle D_j \rangle.$$

We immediately have $H(x_j)^{\Uparrow x_j, s_j+1} \subseteq \mathit{decl}(H) \subseteq N$. Now we show that $\mathit{induc}_{\mathcal{P}}\langle D_j \rangle^{\Uparrow x_j, s_j+1} \subseteq N$. Let $\boldsymbol{f} \in \mathit{induc}_{\mathcal{P}}\langle D_j \rangle$. Let $\varphi \in \mathit{induc}_{\mathcal{P}}$ and $V$ respectively be a rule and valuation that have derived $\boldsymbol{f}$. Let $\varphi'$ be the rule in $\mathit{pure}(\mathcal{P})$ that is obtained after applying transformation (4.4) to $\varphi$. Thus, besides the additional location variable, the rule $\varphi'$ has two timestamp variables, one in the body and one in the head. Moreover, the body contains an additional positive `tsucc`-atom. Let $V'$ be $V$ extended to assign $x_j$ to the location variable, and to assign timestamps $s_j$ and $s_j + 1$ to the body and head timestamp variables respectively. Let $\psi$ be the *positive* ground rule obtained from $\varphi'$ by applying valuation $V'$ and by subsequently removing all negative (ground) body atoms. We show that $\psi \in \mathit{gr}_M^{\mathcal{P},H}$ and that its body is in $N$, so that $\psi$ derives $\mathit{head}_\psi = \boldsymbol{f}^{\Uparrow x_j, s_j+1} \in N$, as desired.

- In order for $\psi$ to be in $\mathit{gr}_M^{\mathcal{P},H}$, it is required that $V'(\mathit{neg}_{\varphi'}) \cap M = \emptyset$. Because $V$ is satisfying for $\varphi$, we have $V(\mathit{neg}_\varphi) \cap D_j = \emptyset$. By Claim A.2 we then have $V(\mathit{neg}_\varphi)^{\Uparrow x_j, s_j} \cap M = \emptyset$. And since $V'(\mathit{neg}_{\varphi'}) = V(\mathit{neg}_\varphi)^{\Uparrow x_j, s_j}$, we have $V'(\mathit{neg}_{\varphi'}) \cap M = \emptyset$, as desired.
- Now we show that $\mathit{pos}_\psi \subseteq N$. Since $\mathit{pos}_\psi = V'(\mathit{pos}_{\varphi'})$, we will show that $V'(\mathit{pos}_{\varphi'}) \subseteq N$. The set $V'(\mathit{pos}_{\varphi'})$ consists of the facts $V(\mathit{pos}_\varphi)^{\Uparrow x_j, s_j}$ and the fact $\mathtt{tsucc}(s_j, s_j+1)$. The latter fact is in $\mathit{decl}(H)$ and thus in $N$. For the other facts, because $V$ is satisfying for $\varphi$, we have $V(\mathit{pos}_\varphi) \subseteq D_j$ and thus $V(\mathit{pos}_\varphi)^{\Uparrow x_j, s_j} \subseteq D_j^{\Uparrow x_j, s_j} = \mathrm{duc}_{\mathcal{R}}^{[j]}$. And by using the given assumption $\mathrm{duc}_{\mathcal{R}}^{[j]} \subseteq N$, we obtain the desired result.

□

**Claim A.4.** Let $i$ be a transition. Suppose that $D_i^{\Uparrow x_i, s_i} \subseteq N$. For each $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and timestamp $t \in \mathbb{N}$ with $(y, t) \not\prec_{\mathcal{R}} (x_i, s_i)$ we have

$$\mathtt{cand}_R(x_i, s_i, y, t, \bar{a}) \in N.$$

*Proof.* By definition of $\mathrm{mesg}_{\mathcal{R}}^{[i]}$, we have $R(y, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. Let $\varphi \in \mathit{async}_{\mathcal{P}}$ and $V$ be a rule and valuation that produced $R(y, \bar{a})$. Let $\varphi' \in \mathcal{P}$ be the original asynchronous rule on which $\varphi$ is based. Let $\varphi'' \in \mathit{pure}(\mathcal{P})$ be the rule obtained from $\varphi'$ by applying transformation (4.5). Let $V''$ be valuation $V$ extended to

assign $x_i$ and $s_i$ to respectively the sender location and sender timestamp of $\varphi''$, and to assign $y$ and $t$ respectively to the addressee location and addressee arrival timestamp. Let $\psi$ denote the *positive* ground rule that is obtained from $\varphi''$ by applying valuation $V''$ and by subsequently removing all negative (ground) body atoms. We show that $\psi \in gr_M^{\mathcal{P},H}$ and that its body is in $N$, so that $\psi$ derives $head_\psi = \texttt{cand}_R(x_i, s_i, y, t, \bar{a}) \in N$, as desired.

- In order for $\psi$ to be in $gr_M^{\mathcal{P},H}$, it is required that $V''(neg_{\varphi''}) \cap M = \emptyset$. By construction of $\varphi''$, the set $V''(neg_{\varphi''})$ consists of the facts $V(neg_\varphi)^{\Uparrow x_i, s_i}$ and the fact $\texttt{before}(y, t, x_i, s_i)$. First, because $V$ is satisfying for $\varphi$, we have $V(neg_\varphi) \cap D_i = \emptyset$. Then, by Claim A.2, we have $V(neg_\varphi)^{\Uparrow x_i, s_i} \cap M = \emptyset$. Moreover, we are given that $(y, t) \not\prec_\mathcal{R} (x_i, s_i)$, and thus we have not added $\texttt{before}(y, t, x_i, s_i)$ to $\text{caus}_\mathcal{R}^{[i]}$, and by extension also not to $M$ (since $\text{caus}_\mathcal{R}^{[i]}$ is the only part of $M$ where we add `before`-facts with last two components $x_i$ and $s_i$). Thus overall $V''(neg_{\varphi''}) \cap M = \emptyset$, as desired.
- Now we show that $pos_\psi \subseteq N$. Since $pos_\psi = V''(pos_{\varphi''})$, we will show that $V''(pos_{\varphi''}) \subseteq N$. By construction of $\varphi''$, the set $V''(pos_{\varphi''})$ consists of the facts $V(pos_\varphi)^{\Uparrow x_i, s_i}$, $\texttt{all}(y)$ and $\texttt{time}(t)$. First, we immediately have $\texttt{time}(t) \in decl(H) \subseteq N$. Also, by definition of $\text{mesg}_\mathcal{R}^{[i]}$, $y$ is a valid addressee and thus $\texttt{all}(y) \in decl(H) \subseteq N$. Finally, because $V$ is satisfying for $\varphi$, we have $V(pos_\varphi) \subseteq D_i$. Thus $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq D_i^{\Uparrow x_i, s_i}$, and we are given that $D_i^{\Uparrow x_i, s_i} \subseteq N$. Thus overall $V''(pos_{\varphi''}) \subseteq N$.

□

## A.2 Second Inclusion

In this section we show that $N \subseteq M$. By definition, $N = gr_M^{\mathcal{P},H}(decl(H))$. Following the semantics of positive Datalog$^\neg$ programs in Section 2.2.1, we can view $N$ as a fixpoint, i.e., $N = \bigcup_{l \in \mathbb{N}} N_l$, where $N_0 = decl(H)$, and for each $l \geq 1$ the set $N_l$ is obtained by applying the immediate consequence operator of $gr_M^{\mathcal{P},H}$ to $N_{l-1}$. This implies $N_{l-1} \subseteq N_l$ for each $l \geq 1$. We show by induction on $l = 0, 1, \ldots$, that $N_l \subseteq M$. For the base case ($l = 0$), we immediately have $N_0 = decl(H) \subseteq M$. For the induction hypothesis, we assume for some $l \geq 1$ that $N_{l-1} \subseteq M$. For the inductive step, we show that $N_l \subseteq N$. Specifically, we divide the facts of $N_l \setminus N_{l-1}$ into groups based on their predicate, and for each group we show inclusion in $M$. As for terminology, we call a ground rule $\psi$ *active* on $N_{l-1}$ if $pos_\psi \subseteq N_{l-1}$.

The numbered claims we will refer to can be found in Section A.2.1.

**Causality** Let $\texttt{before}(x, s, y, t) \in N_l \setminus N_{l-1}$. It is sufficient to show that $(x, s) \prec_\mathcal{R} (y, t)$ because then $\texttt{before}(x, s, y, t) \in \text{caus}_\mathcal{R}^{[i]} \subseteq M$ where $i = glob_\mathcal{R}(y, t)$. We have the following cases:

- The `before`-fact was derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.1) (local edge). This implies $x = y$ and $t = s + 1$. We immediately have $(x, s) \prec_{\mathcal{R}} (y, t)$.

- The `before`-fact was derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.9) (message edge):

$$\texttt{before}(x, s, y, t) \leftarrow \texttt{chosen}_R(x, s, y, t, \bar{a}).$$

  Since this rule is active on $N_{l-1}$, we have $\texttt{chosen}_R(x, s, y, t, \bar{a}) \in N_{l-1}$. By applying the induction hypothesis, we have $\texttt{chosen}_R(x, s, y, t, \bar{a}) \in M$. Denoting $j = glob_{\mathcal{R}}(x, s)$, the set $\text{snd}_{\mathcal{R}}^{[j]}$ is the only part of $M$ where we could have added this fact. This implies that $x$ during its step $s$ sends a message to $y$, and this message arrives at local step $t$ of $y$. Therefore $(x, s) \prec_{\mathcal{R}} (y, t)$, as desired.

- The `before`-fact was derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.2) (transitive edge):

$$\texttt{before}(x, s, y, t) \leftarrow \texttt{before}(x, s, z, u), \texttt{before}(z, u, y, t).$$

  Since this rule is active on $N_{l-1}$, its body facts are in $N_{l-1}$. By applying the induction hypothesis, we have $\texttt{before}(x, s, z, u) \in M$ and $\texttt{before}(z, u, y, t) \in M$. The only places we could have added these facts to $M$ are in the sets $\text{caus}_{\mathcal{R}}^{[j]}$ and $\text{caus}_{\mathcal{R}}^{[k]}$ respectively, where $j = glob_{\mathcal{R}}(z, u)$ and $k = glob_{\mathcal{R}}(y, t)$. By construction of the sets $\text{caus}_{\mathcal{R}}^{[j]}$ and $\text{caus}_{\mathcal{R}}^{[k]}$ we respectively have that $(x, s) \prec_{\mathcal{R}} (z, u)$ and $(z, u) \prec_{\mathcal{R}} (y, t)$, and thus by transitivity $(x, s) \prec_{\mathcal{R}} (y, t)$, as desired.

**Finite Messages**

- Let $\texttt{hasSender}(x, s, y, t) \in N_l \setminus N_{l-1}$. This fact can only have been derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.10):

$$\texttt{hasSender}(x, s, y, t) \leftarrow \texttt{chosen}_R(y, t, x, s, \bar{a}).$$

  Since this rule is active on $N_{l-1}$, we have $\texttt{chosen}_R(y, t, x, s, \bar{a}) \in N_{l-1}$. By applying the induction hypothesis, we have $\texttt{chosen}_R(y, t, x, s, \bar{a}) \in M$. We can only have added this fact in the set $\text{snd}_{\mathcal{R}}^{[i]}$ with $i = glob_{\mathcal{R}}(y, t)$. This means that $y$ during its step $t$ sends a message $R(\bar{a})$ to $x$, and this message arrives during step $s$ of $x$. Hence, denoting $j = glob_{\mathcal{R}}(x, s)$, we have $(y, t) \in \text{senders}_{\mathcal{R}}^{[j]}$ (with $\text{senders}_{\mathcal{R}}^{[j]}$ as defined in Section 6). Thus we have added the fact $\texttt{hasSender}(x, s, y, t) \in \text{fin}_{\mathcal{R}}^{[j]} \subseteq M$, as desired.

- Let $\texttt{isSmaller}(x, s, y, t) \in N_l \setminus N_{l-1}$. This fact can only have been derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.11):

$$\texttt{isSmaller}(x, s, y, t) \leftarrow \texttt{hasSender}(x, s, y, t), \texttt{hasSender}(x, s, y, t'), \; t < t'.$$

Since this rule is active on $N_{l-1}$, its body facts are in $N_{l-1}$. By applying the induction hypothesis, we have $\texttt{hasSender}(x, s, y, t) \in M$ and $\texttt{hasSender}(x, s, y, t') \in M$. The only part of $M$ where we could have added these facts is the set $\text{fin}_{\mathcal{R}}^{[i]}$ with $i = \mathit{glob}_{\mathcal{R}}(x, s)$. By construction of the set $\text{fin}_{\mathcal{R}}^{[i]}$, this implies that $(y, t) \in \text{senders}_{\mathcal{R}}^{[i]}$ and $(y, t') \in \text{senders}_{\mathcal{R}}^{[i]}$. Because $(t < t') \in N_{l-1}$, we more specifically know that $(t < t') \in \mathit{decl}(H)$, which implies $t < t'$. Thus we have added $\texttt{isSmaller}(x, s, y, t) \in \text{fin}_{\mathcal{R}}^{[i]}$, as desired.

- Let $\texttt{hasMax}(x, s, y) \in N_l \setminus N_{l-1}$. This fact can only have been derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.12):

$$\texttt{hasMax}(x, s, y) \leftarrow \texttt{hasSender}(x, s, y, t).$$

  Since this rule is active on $N_{l-1}$, we have $\texttt{hasSender}(x, s, y, t) \in N_{l-1}$. By applying the induction hypothesis, we have $\texttt{hasSender}(x, s, y, t) \in M$. The only part of $M$ where we could have added this fact, is the set $\text{fin}_{\mathcal{R}}^{[i]}$ with $i = \mathit{glob}_{\mathcal{R}}(x, s)$. Thus $(y, t) \in \text{senders}_{\mathcal{R}}^{[i]}$, and $y$ is a sender-node mentioned in $\text{senders}_{\mathcal{R}}^{[i]}$. Hence, we have added $\texttt{hasMax}(x, s, y) \in \text{fin}_{\mathcal{R}}^{[i]} \subseteq M$, as desired.

- Let $\texttt{rcvInf}(x, s) \in N_l \setminus N_{l-1}$. This fact can only have been derived by a ground rule in $gr_M^{\mathcal{P},H}$ of the form (4.13):

$$\texttt{rcvInf}(x, s) \leftarrow \texttt{hasSender}(x, s, y, t).$$

  Since this rule is active on $N_{l-1}$, we have $\texttt{hasSender}(x, s, y, t) \in N_{l-1}$. By applying the induction hypothesis, we have $\texttt{hasSender}(x, s, y, t) \in M$. The only part of $M$ where we could have added this fact, is the set $\text{fin}_{\mathcal{R}}^{[i]}$ with $i = \mathit{glob}_{\mathcal{R}}(x, s)$. Thus $(y, t) \in \text{senders}_{\mathcal{R}}^{[i]}$. Moreover, because the rule (4.13) contains a negative $\texttt{hasMax}$-atom in the body, and the above ground rule is in $gr_M^{\mathcal{P},H}$, it must be that $\texttt{hasMax}(x, s, y) \notin M$, and thus $\texttt{hasMax}(x, s, y) \notin \text{fin}_{\mathcal{R}}^{[i]}$. But since $y$ is a sender-node mentioned in $\text{senders}_{\mathcal{R}}^{[i]}$, the absence of $\texttt{hasMax}(x, s, y)$ from $\text{fin}_{\mathcal{R}}^{[i]}$ is impossible. Therefore this case can not occur.

**Deductive** Let $R(x, s, \bar{a}) \in (N_l \setminus N_{l-1})|_{sch(\mathcal{P})^{\text{LT}}}$. The fact $R(x, s, \bar{a})$ has been derived by a ground rule $\psi \in gr_M^{\mathcal{P},H}$ that is active on $N_{l-1}$. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in \mathit{pure}(\mathcal{P})$ and valuation $V$ such that $\psi$ is obtained from $\varphi$ by applying $V$ and by subsequently removing the negative (ground) body atoms, and such that $V(\mathit{neg}_\varphi) \cap M = \emptyset$. We have the following cases:

- Rule $\varphi$ is of the form (4.3). Let $\varphi' \in \mathit{deduc}_{\mathcal{P}}$ be the original deductive rule corresponding to $\varphi$. By construction of $\varphi$ out of $\varphi'$, we can apply valuation $V$ to $\varphi'$ as well. Denote $i = \mathit{glob}_{\mathcal{R}}(x, s)$. We will show now that $V$ is

satisfying for $\varphi'$ during transition $i$, which causes $V(\mathit{head}_{\varphi'}) = R(\bar{a}) \in D_i$ to be derived, and we obtain as desired:

$$R(x, s, \bar{a}) \in D_i^{\Uparrow x,s} = D_i^{\Uparrow x_i, s_i} = \mathrm{duc}_{\mathcal{R}}^{[i]} \subseteq M.$$

By definition of syntactic stratification, relations mentioned in $\mathit{pos}_{\varphi'}$ are never computed in a stratum higher than $R$, and relations mentioned in $\mathit{neg}_{\varphi'}$ are computed in a strictly lower stratum than $R$. Thus, it is sufficient to show that $V(\mathit{pos}_{\varphi'}) \subseteq D_i$ and $V(\mathit{neg}_{\varphi'}) \cap D_i = \emptyset$. First, we show that $V(\mathit{pos}_{\varphi'}) \subseteq D_i$. Because $\varphi$ is of the form (4.3), all facts in $V(\mathit{pos}_{\varphi})$ are over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$ and have location specifier $x$ and timestamp $s$. Moreover, since $\psi$ is active on $N_{l-1}$, we have $\mathit{pos}_{\psi} = V(\mathit{pos}_{\varphi}) \subseteq N_{l-1}$. By applying the induction hypothesis, we have $V(\mathit{pos}_{\varphi}) \subseteq M$, and thus by Claim B.4 we have $V(\mathit{pos}_{\varphi})^{\Downarrow} \subseteq D_i$. By construction of $\varphi$ out of $\varphi'$, we have $V(\mathit{pos}_{\varphi})^{\Downarrow} = V(\mathit{pos}_{\varphi'})$. Thus $V(\mathit{pos}_{\varphi'}) \subseteq D_i$, as desired.

Secondly, we show that $V(\mathit{neg}_{\varphi'}) \cap D_i = \emptyset$. Because $\varphi$ is of the form (4.3), all facts in $V(\mathit{neg}_{\varphi})$ are over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$ and have location specifier $x$ and timestamp $s$. Moreover, by choice of $\varphi$ and $V$, we have $V(\mathit{neg}_{\varphi}) \cap M = \emptyset$, and thus by Claim B.5 we have $V(\mathit{neg}_{\varphi})^{\Downarrow} \cap D_i = \emptyset$. And by construction of $\varphi$ out of $\varphi'$, we have $V(\mathit{neg}_{\varphi})^{\Downarrow} = V(\mathit{neg}_{\varphi'})$. Thus $V(\mathit{neg}_{\varphi'}) \cap D_i = \emptyset$, as desired.

- Rule $\varphi$ is of the form (4.4). Let $\varphi' \in \mathit{induc}_{\mathcal{P}}$ be the original inductive rule corresponding to $\varphi$. First, $\psi$ contains in its body a fact of the form $\mathtt{tsucc}(r, s)$. Since $\psi$ is active on $N_{l-1}$, we have $\mathtt{tsucc}(r, s) \in N_{l-1}$ and more specifically, $\mathtt{tsucc}(r, s) \in \mathit{decl}(H)$. This implies that $s = r + 1$. Denote $i = \mathit{glob}_{\mathcal{R}}(x, r)$ and $j = \mathit{glob}_{\mathcal{R}}(x, s)$. Note that the relationship between timestamps $r$ and $s$ implies that there are no transitions of node $x$ between $i$ and $j$. By construction of $\varphi$ out of rule $\varphi'$, we can apply $V$ to $\varphi'$, and we will now show that $V$ is satisfying for $\varphi'$ during transition $i$. This results in $V(\mathit{head}_{\varphi'}) = R(\bar{a}) \in \mathit{induc}_{\mathcal{P}}\langle D_i \rangle \subseteq \mathrm{st}^{\rho_{i+1}}(x)$, and since $\mathrm{st}^{\rho_{i+1}}(x) = \mathrm{st}^{\rho_j}(x) \subseteq D_j$, we obtain $R(x, s, \bar{a}) \in D_j^{\Uparrow x,s} = \mathrm{duc}_{\mathcal{R}}^{[j]} \subseteq M$, as desired.

  We have to show that $V(\mathit{pos}_{\varphi'}) \subseteq D_i$ and $V(\mathit{neg}_{\varphi'}) \cap D_i = \emptyset$. First, we show that $V(\mathit{pos}_{\varphi'}) \subseteq D_i$. Denote $I = V(\mathit{pos}_{\varphi})|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}}$, which allows us to exclude the extra $\mathtt{tsucc}$-fact in the body. All facts in $I$ have location specifier $x$ and timestamp $s$. Because $\psi$ is active on $N_{l-1}$, we have $I \subseteq \mathit{pos}_{\psi} \subseteq N_{l-1}$, and by applying the induction hypothesis, we have $I \subseteq M$. By Claim B.4 thus have $I^{\Downarrow} \subseteq D_i$, and since $I^{\Downarrow} = V(\mathit{pos}_{\varphi'})$, we have $V(\mathit{pos}_{\varphi'}) \subseteq D_i$, as desired.

  Secondly, showing that $V(\mathit{neg}_{\varphi'}) \cap D_i = \emptyset$ is like in the previous case, where $\varphi$ is deductive.

- Rule $\varphi$ is of the form (4.8). Then $\psi$ concretely looks as follows, where $(y, t) \in \mathcal{N} \times \mathbb{N}$ and $R$ in $\mathit{idb}(\mathcal{P})$:

$$R(x, s, \bar{a}) \leftarrow \mathtt{chosen}_R(y, t, x, s, \bar{a}).$$

Since $\psi$ is active on $N_{l-1}$, we have $\texttt{chosen}_R(y,t,x,s,\bar{a}) \in N_{l-1}$, and by applying the induction hypothesis, we have $\texttt{chosen}_R(y,t,x,s,\bar{a}) \in M$. The only part of $M$ where we could have added this fact is in the set $\text{snd}_{\mathcal{R}}^{[i]}$ with $i = \mathit{glob}_{\mathcal{R}}(y,t)$. This implies that $R(x,\bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$ and that node $x$ will receive fact $R(\bar{a})$ during its local step $s$, thus during transition $j = \mathit{glob}_{\mathcal{R}}(x,s)$. Then, by the operational semantics, we have $R(\bar{a}) \in \mathit{untag}(m_j) \subseteq D_j$. Thus $R(x,s,\bar{a}) \in D_j^{\Uparrow x,s} = \text{duc}_{\mathcal{R}}^{[j]} \subseteq M$, as desired.

**Sending**

- Let $\texttt{cand}_R(x,s,y,t,\bar{a}) \in N_l \setminus N_{l-1}$. The fact $\texttt{cand}_R(x,s,y,t,\bar{a})$ is derived by a ground rule $\psi \in gr_M^{\mathcal{P},H}$ of the form (4.5) that is active on $N_{l-1}$. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in \mathit{pure}(\mathcal{P})$ and a valuation $V$ such that $\psi$ is obtained from $\varphi$ by applying valuation $V$ and by subsequently removing the negative (ground) body atoms, and so that $V(\mathit{neg}_\varphi) \cap M = \emptyset$. Denote $i = \mathit{glob}_{\mathcal{R}}(x,s)$. It is sufficient to show that $R(y,\bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$ and $(y,t) \not\prec_{\mathcal{R}} (x,s)$, because then $\texttt{cand}_R(x,s,y,t,\bar{a}) \in \text{snd}_{\mathcal{R}}^{[i]} \subseteq M$, as desired.

  First, we show that $(y,t) \not\prec_{\mathcal{R}} (x,s)$. Because there is a negative $\texttt{before}$-atom in $\varphi$, the existence of $\psi$ in $gr_M^{\mathcal{P},H}$ implies that $\texttt{before}(y,t,x,s) \notin M$. Hence, $\texttt{before}(y,t,x,s) \notin \text{caus}_{\mathcal{R}}^{[i]}$. Then by construction of $\text{caus}_{\mathcal{R}}^{[i]}$ we obtain $(y,t) \not\prec_{\mathcal{R}} (x,s)$.

  Secondly, we show that $R(y,\bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$. Let $\varphi' \in \mathcal{P}$ be the original asynchronous rule on which $\varphi$ is based. Let $\varphi'' \in \mathit{async}_{\mathcal{P}}$ be the rule corresponding to $\varphi'$. It follows from the constructions of $\varphi$ out of $\varphi'$ and $\varphi''$ out of $\varphi'$ that valuation $V$ can be applied to $\varphi''$. We have $V(\mathit{head}_{\varphi''}) = R(y,\bar{a})$. We will show that $V$ is satisfying for $\varphi''$ during transition $i$, which gives $R(y,\bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. Moreover, the body of $\psi$ contains the fact $\texttt{all}(y) \in \mathit{decl}(H)$, and thus $y \in \mathcal{N}$, making $y$ a valid addressee. Thus $R(y,\bar{a}) \in \text{mesg}_{\mathcal{R}}^{[i]}$, as desired.

  We have to show that $V(\mathit{pos}_{\varphi''}) \subseteq D_i$ and $V(\mathit{neg}_{\varphi''}) \cap D_i = \emptyset$. Abbreviate $I_1 = V(\mathit{pos}_\varphi)|_{\mathit{sch}(\mathcal{P})^{\text{LT}}}$ and $I_2 = V(\mathit{neg}_\varphi)|_{\mathit{sch}(\mathcal{P})^{\text{LT}}}$. Note that $I_1^{\Downarrow} = V(\mathit{pos}_{\varphi''})$ and $I_2^{\Downarrow} = V(\mathit{neg}_{\varphi''})$. All facts in $I_1 \cup I_2$ have location specifier $x$ and timestamp $s$.

  - Because $\psi$ is active on $N_{l-1}$, we have $I_1 \subseteq \mathit{pos}_\psi \subseteq N_{l-1}$, and by applying the induction hypothesis we thus have $I_1 \subseteq M$. Then by Claim B.4 we have $I_1^{\Downarrow} \subseteq D_i$ and thus $V(\mathit{pos}_{\varphi''}) \subseteq D_i$, as desired.
  - By choice of $\varphi$ and $V$, we have $I_2 \cap M = \emptyset$. By Claim B.5 we then have $I_2^{\Downarrow} \cap D_i = \emptyset$ and thus $V(\mathit{neg}_{\varphi''}) \cap D_i = \emptyset$, as desired.

- Let $\texttt{chosen}_R(x,s,y,t,\bar{a}) \in N_l \setminus N_{l-1}$. This fact is derived by a ground rule $\psi$ in $gr_M^{\mathcal{P},H}$ of the form (4.6):

$$\texttt{chosen}_R(x,s,y,t,\bar{a}) \leftarrow \texttt{cand}_R(x,s,y,t,\bar{a}).$$

Denote $i = \mathit{glob}_{\mathcal{R}}(x, s)$. We will show that $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and that $t$ is the actual arrival timestamp of this message at $y$. Then we will have added $\mathtt{chosen}_R(x, s, y, t, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]} \subseteq M$, as desired.

First, since $\psi$ is active on $N_{l-1}$, we have $\mathtt{cand}_R(x, s, y, t, \bar{a}) \in N_{l-1}$, and by applying the induction hypothesis, we have $\mathtt{cand}_R(x, s, y, t, \bar{a}) \in M$. The set $\mathrm{snd}_{\mathcal{R}}^{[i]}$ is the only part of $M$ where we could have added this fact. Next, $\mathtt{cand}_R(x, s, y, t, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]}$ implies $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and $(y, t) \not\prec_{\mathcal{R}} (x, s)$.

We are left to show that $t$ is the actual arrival timestamp of the message. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in \mathit{pure}(\mathcal{P})$ and valuation $V$ such that $\psi$ is obtained from $\varphi$ by applying $V$ and by subsequently removing the negative (ground) body atoms, and so that $V(\mathit{neg}_\varphi) \cap M = \emptyset$. Now, because rule $\varphi$ contains a negative $\mathtt{other}_R$-atom in its body, we have $\mathtt{other}_R(x, s, y, t, \bar{a}) \notin M$ and thus $\mathtt{other}_R(x, s, y, t, \bar{a}) \notin \mathrm{snd}_{\mathcal{R}}^{[i]}$. Since $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and $(y, t) \not\prec_{\mathcal{R}} (x, s)$ (see above), the absence of this $\mathtt{other}_R$-fact from $\mathrm{snd}_{\mathcal{R}}^{[i]}$ can only be explained by the following: $t = \mathit{loc}_{\mathcal{R}}(j)$ with $j = \alpha_{\mathcal{R}}(i, y, R(\bar{a}))$, as desired.

- Let $\mathtt{other}_R(x, s, y, t, \bar{a}) \in N_l \setminus N_{l-1}$. This fact is derived by a ground rule $\psi$ of the form (4.7):

$$\mathtt{other}_R(x, s, y, t, \bar{a}) \quad \leftarrow \quad \mathtt{cand}_R(x, s, y, t, \bar{a}),\ \mathtt{chosen}_R(x, s, y, t', \bar{a}),\ t \neq t'.$$

  Since this rule is active on $N_{l-1}$, we have $\mathtt{cand}_R(x, s, y, t, \bar{a}) \in N_{l-1}$ and $\mathtt{chosen}_R(x, s, y, t', \bar{a}) \in N_{l-1}$, and these facts are also in $M$ by applying the induction hypothesis. Denote $i = \mathit{glob}_{\mathcal{R}}(x, s)$. The only part of $M$ where we could have added these fact to $M$, is the set $\mathrm{snd}_{\mathcal{R}}^{[i]}$. First, $\mathtt{cand}_R(x, s, y, t, \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]}$ implies that $R(y, \bar{a}) \in \mathrm{mesg}_{\mathcal{R}}^{[i]}$ and $(y, t) \not\prec_{\mathcal{R}} (x, s)$. Second, $\mathtt{chosen}_R(x, s, y, t', \bar{a}) \in \mathrm{snd}_{\mathcal{R}}^{[i]}$ implies that $t'$ is the real arrival timestamp of the message $R(\bar{a})$ at $y$. Finally, since $\psi$ is active, we have $(t \neq t') \in \mathit{decl}(H)$, and thus $t \neq t'$. Therefore we have added $\mathtt{other}_R(x, s, y, t, \bar{a})$ to $\mathrm{snd}_{\mathcal{R}}^{[i]} \subseteq M$, as desired.

### A.2.1 Subclaims

**Claim A.5.** Let $I$ be a set of facts over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$, all having the same location specifier $x \in \mathcal{N}$ and timestamp $s \in \mathbb{N}$. Denote $i = \mathit{glob}_{\mathcal{R}}(x, s)$. If $I \cap M = \emptyset$ then $I^{\Downarrow} \cap D_i = \emptyset$.

*Proof.* First, $I \cap M = \emptyset$ implies $I \cap \mathrm{duc}_{\mathcal{R}}^{[i]} = \emptyset$ because $\mathrm{duc}_{\mathcal{R}}^{[i]} \subseteq M$. And since $\mathrm{duc}_{\mathcal{R}}^{[i]} = D_i^{\Uparrow x,s}$, we have $I \cap D_i^{\Uparrow x,s} = \emptyset$. Finally, since the facts in $I \cup D_i^{\Uparrow x,s}$ all have the same location specifier $x$ and timestamp $s$, we obtain $I^{\Downarrow} \cap D_i = \emptyset$. □

**Claim A.6.** Let $I$ be a set of facts over $sch(\mathcal{P})^{\mathrm{LT}}$, all having the same location specifier $x \in \mathcal{N}$ and timestamp $s \in \mathbb{N}$. Denote $i = glob_{\mathcal{R}}(x, s)$. If $I \subseteq M$ then $I^{\Downarrow} \subseteq D_i$.

*Proof.* The only part of $M$ where we add facts over $sch(\mathcal{P})^{\mathrm{LT}}$ with location specifier $x$ and timestamp $s$ is $\mathrm{duc}_{\mathcal{R}}^{[i]}$. Hence $I \subseteq \mathrm{duc}_{\mathcal{R}}^{[i]} = D_i^{\Uparrow x,s}$ and thus $I^{\Downarrow} \subseteq D_i$. □

# B Model to Run: Proof Details

Consider the definitions and notations from Section 7. In this section we show that $\mathcal{R}$ is a run of $\mathcal{P}$ on input $H$, such that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\mathrm{LT}}}$. We do this in several parts, where each part is placed in its own subsection:

- in Section B.2 we show that $\rho_0 = start(\mathcal{P}, H)$;
- in Section B.3 we show that every fabricated transition of $\mathcal{R}$ is in fact a valid transition; and,
- in Section B.4 we show that $trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\mathrm{LT}}}$.

Before we start, we need some additional notations that are given in the next subsection. The numbered claims we will refer to can be found in Section B.5.

## B.1 Additional Definitions and Notations

Let $\varphi \in pure(\mathcal{P})$ be a rule such that its head atom is over $sch(\mathcal{P})^{\mathrm{LT}}$. From the construction of $pure(\mathcal{P})$, it is clear that $\varphi$ falls in exactly one of the following three cases:

- $\varphi$ is of the form (4.3), i.e., *deductive*, recognizable as a rule in which only atoms over $sch(\mathcal{P})^{\mathrm{LT}}$ are used, and in which the location and timestamp variable in the head are the same as in the body;
- $\varphi$ is of the form (4.4), i.e., *inductive*, recognizable as a rule with a head atom over $sch(\mathcal{P})^{\mathrm{LT}}$ and a `tsucc`-atom in the body;
- $\varphi$ is of the form (4.8), i.e., a *delivery*, recognizable as a rule with a head atom over $sch(\mathcal{P})^{\mathrm{LT}}$ and a $\texttt{chosen}_R$-fact in the body (where $R$ is also the head-predicate).

The same classification of deductive, inductive and delivery rules can also be applied to the (positive) ground rules in $gr_M^{\mathcal{P},H}$ that have a ground head atom over $sch(\mathcal{P})^{\mathrm{LT}}$.

Recall from the general remarks in the appendix that we are working with a fixed (but arbitrary) syntactic stratification for the deductive rules. Stratum numbers start at 1. If $\varphi \in pure(\mathcal{P})$ is deductive, we can uniquely identify its stratum number as the stratum number of the original deductive rule in $\mathcal{P}$ on

which $\varphi$ is based. Similarly, for deductive ground rules, we can also uniquely identify the stratum number as the stratum number of a corresponding non-ground rule in $\mathit{pure}(\mathcal{P})$.[9]

We call a ground rule $\psi \in \mathit{gr}_M^{\mathcal{P},H}$ *active* if $\mathit{pos}_\psi \subseteq M$, which implies that $\mathit{head}_\psi \in M$ because $M$ is stable. Now we define the following subsets of $M$:

- $M^{\mathrm{duc},k}$: all facts of $M$ that are the head of an active deductive rule in $\mathit{gr}_M^{\mathcal{P},H}$ with stratum number less than or equal to $k$;
- $M^{\mathrm{ind}}$: all facts of $M$ that are the head of an active inductive rule in $\mathit{gr}_M^{\mathcal{P},H}$;
- $M^{\mathrm{deliv}}$: all facts of $M$ that are the head of an active delivery rule in $\mathit{gr}_M^{\mathcal{P},H}$.

This allows us to classify the facts in $M|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}}$ as being derived in a deductive manner, an inductive manner or being message deliveries. We also define:

$$M^{\blacktriangle} = M|_{\mathit{edb}(\mathcal{P})^{\mathrm{LT}}} \cup M^{\mathrm{ind}} \cup M^{\mathrm{deliv}}.$$

For $(x, s) \in \mathcal{N} \times \mathbb{N}$, we write $I|^{x,s}$ to abbreviate $(I|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}})|^{x,s}$. So intuitively, when we select the facts with location specifier $x$ and timestamp $s$, we are only interested in facts that provide these two components, which are the facts over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$.

For $i \in \mathbb{N}$, the set $(M^{\blacktriangle})|^{x_i,s_i}$ intuitively contains the "input" of the deductive fixpoint computation during the local step $s_i$ of the node $x_i$: the input facts, the facts derived by inductive rules during a previous computation step (if any) of $x_i$, and the delivered messages. The deductive rules then complete this information by deriving some new facts, that are visible within step $s_i$ of $x_i$.

For a transition number $i$ of $\mathcal{R}$, we denote $D_i = \mathit{deduc}_{\mathcal{P}}(\mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i))$. For a number $k \in \mathbb{N}$, we write $D_i^{\to k}$ to denote the set of facts obtained by adding to $\mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i)$ all facts derived in stratum 1 up to stratum $k$ during the computation of $D_i$. To mirror this notation, we write $M^{\to k}$ to denote the set $M^{\blacktriangle} \cup M^{\mathrm{duc},k}$. For uniformity in the proofs, we will consider the case $k = 0$, which is an invalid stratum number, and this gives $D_i^{\to 0} = \mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i)$ and $M^{\to 0} = M^{\blacktriangle}$.

## B.2 Valid Start

We show that $\rho_0 = \mathit{start}(\mathcal{P}, H)$. Let $x \in \mathcal{N}$. First we show that $\mathrm{st}^{\rho_0}(x)$ is the correct start state for $x$, meaning that $\mathrm{st}^{\rho_0}(x) = H(x)$. By definition,

$$\mathrm{st}^{\rho_0}(x) = \left((M|_{\mathit{edb}(\mathcal{P})^{\mathrm{LT}}})|^{x,s} \cup M^{\mathrm{ind}}|^{x,s}\right)^{\Downarrow}$$

with $s = \mathit{loc}_M(0, x)$. It must be that $s = 0$ because there are no pairs of $\mathcal{N} \times \mathbb{N}$ with first component $x$ that have an ordinal strictly less than 0. Now, there can

[9] We say *a* rather than *the* corresponding rule because there could be more than one. Indeed, multiple original deductive rules in $\mathit{pure}(\mathcal{P})$ could be mapped to the same positive ground rule after applying a valuation and removing their negative ground body atoms. But in any case, these non-ground rules will have the same head predicate. Hence, they have the same stratum.

be no ground inductive rules in $gr_M^{\mathcal{P},H}$ that derive facts with head timestamp 0 because it follows from the construction of $decl(H)$ that the second component of a `tsucc`-fact is always strictly larger than 0. Therefore $M^{\text{ind}}|^{x,s} = \emptyset$, and thus $\text{st}^{\rho_0}(x) = \left((M|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s}\right)^{\Downarrow}$. Then by Claim B.1 we have $\text{st}^{\rho_0}(x) = (H(x)^{\Uparrow x,s})^{\Downarrow} = H(x)$, as desired.

Now we show that $\text{bf}^{\rho_0}(x)$ is the correct start message buffer for $x$, which means that it should be empty. By definition, $\text{bf}^{\rho_0}(x)$ is

$$\{\langle glob_M(y,t),\, R(\bar{a})\rangle \mid \\ \exists u : \texttt{chosen}_R(y,t,x,u,\bar{a}) \in M,\, glob_M(y,t) < 0 \leq glob_M(x,u)\}.$$

We can immediately see that there are no facts of the form $\texttt{chosen}_R(y,t,x,u,\bar{a}) \in M$ with $glob_M(y,t) < 0$, by definition of the function $glob_M(\cdot)$. Hence, $\text{bf}^{\rho_0}(x) = \emptyset$.

We conclude that $\rho_0 = start(\mathcal{P}, H)$.

## B.3 Valid Transition

Let $i \in \mathbb{N}$. We will now show that $\rho_i \xrightarrow[i]{x_i, m_i} \rho_{i+1}$ is a valid transition.

We start by showing that $m_i \subseteq \text{bf}^{\rho_i}(x_i)$. Let $\langle j, \boldsymbol{f}\rangle \in m_i$. By definition of $m_i$, there is a fact of the form $\texttt{chosen}_R(y,t,z,u,\bar{a}) \in M$ with $glob_M(z,u) = i$ such that $j = glob_M(y,t)$ and $\boldsymbol{f} = R(\bar{a})$. Note that $glob_M(z,u) = i$ implies $z = x_i$ and $u = s_i$. Now, because rules in $pure(\mathcal{P})$ of the form (4.9) are always positive, the following ground rule is in $gr_M^{\mathcal{P},H}$, which is of the form (4.9):

$$\texttt{before}(y,t,x_i,s_i) \leftarrow \texttt{chosen}_R(y,t,x_i,s_i,\bar{a}).$$

Since its body is in $M$, this rule derives $\texttt{before}(y,t,x_i,s_i) \in M$. Hence $(y,t) \prec_M (x_i,s_i)$ by definition of $\prec_M$. Moreover, $<_M$ respects $\prec_M$, and thus $(y,t) <_M (x_i,s_i)$ and $glob_M(y,t) < glob_M(x_i,s_i)$. And since $glob_M(x_i,s_i) = i$, we overall have

$$glob_M(y,t) < i \leq glob_M(x_i,s_i).$$

Therefore $\langle j, \boldsymbol{f}\rangle \in \text{bf}^{\rho_i}(x_i)$.

Now, because $m_i \subseteq \text{bf}^{\rho_i}(x_i)$, and because transitions are deterministic once the active node and delivered messages are fixed, we can consider the unique result configuration $\rho$ such that $\rho_i \xrightarrow[i]{x_i, m_i} \rho$ is a valid transition. We have to show that $\rho_{i+1} = \rho$. We divide the work in two parts: for each $x \in \mathcal{N}$, show that $\text{st}^{\rho_{i+1}}(x) = \text{st}^{\rho}(x)$ (state), and $\text{bf}^{\rho_{i+1}}(x) = \text{bf}^{\rho}(x)$ (buffer).

### B.3.1 State

Let $x \in \mathcal{N}$. We must show that $\text{st}^{\rho_{i+1}}(x) = \text{st}^{\rho}(x)$. Denote $s = loc_M(i+1, x)$. By definition,

$$\text{st}^{\rho_{i+1}}(x) = \left((M|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s} \cup M^{\text{ind}}|^{x,s}\right)^{\Downarrow}.$$

**Case $x \neq x_i$.** By definition, $\mathrm{st}^{\rho}(x) = \mathrm{st}^{\rho_i}(x)$. We will show that $\mathrm{st}^{\rho_{i+1}}(x) = \mathrm{st}^{\rho_i}(x)$, which gives $\mathrm{st}^{\rho_{i+1}}(x) = \mathrm{st}^{\rho}(x)$, as desired. Since $x \neq x_i$, the number of pairs from $\mathcal{N} \times \mathbb{N}$ containing node $x$ that come strictly before ordinal $i+1$ is the same as the number of pairs containing node $x$ that come strictly before ordinal $i$. Formally: $s = \mathit{loc}_M(i+1, x) = \mathit{loc}_M(i, x)$. Thus the right-hand side in the previous equation equals $\mathrm{st}^{\rho_i}(x)$, and the result is obtained.

**Case $x = x_i$.** By definition, $\mathrm{st}^{\rho}(x) = H(x) \cup \mathit{induc}_{\mathcal{P}}\langle D_i \rangle$. Referring to the definition of $\mathrm{st}^{\rho_{i+1}}(x)$ from above, by Claim B.1 we have

$$(M|_{\mathit{edb}(\mathcal{P})^{\mathrm{LT}}})|^{x,s} = H(x)^{\Uparrow x,s}.$$

If we can also show that $M^{\mathrm{ind}}|^{x,s} = \mathit{induc}_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x,s}$, then we overall have, as desired:

$$\begin{aligned} \mathrm{st}^{\rho_{i+1}}(x) &= \left((M|_{\mathit{edb}(\mathcal{P})^{\mathrm{LT}}})|^{x,s} \cup M^{\mathrm{ind}}|^{x,s}\right)^{\Downarrow} \\ &= H(x) \cup \mathit{induc}_{\mathcal{P}}\langle D_i \rangle \\ &= \mathrm{st}^{\rho}(x). \end{aligned}$$

Since $x = x_i$, we have $s = \mathit{loc}_M(i+1, x_i) = \mathit{loc}_M(i, x_i) + 1$, and using that $\mathit{loc}_M(i, x_i) = s_i$ (Claim B.2), we have $s = s_i + 1$. We show that $M^{\mathrm{ind}}|^{x_i, s_i+1} = \mathit{induc}_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x_i, s_i+1}$. Both inclusions are shown separately, by Claims B.3 and B.6. Therefore, we obtain $M^{\mathrm{ind}}|^{x,s} = \mathit{induc}_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x,s}$, as desired.

### B.3.2 Buffer

Let $x \in \mathcal{N}$. We must show that $\mathrm{bf}^{\rho_{i+1}}(x) = \mathrm{bf}^{\rho}(x)$. Denote

$$\delta^{i \to x} = \{\langle i, R(\bar{a}) \rangle \mid R(x, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle\}.$$

Like in the operational semantics, $\delta^{i \to x}$ denotes the (tagged) messages that are sent to $x$ during transition $i$.

**Case $x \neq x_i$.** By definition, $\mathrm{bf}^{\rho}(x) = \mathrm{bf}^{\rho_i}(x) \cup \delta^{i \to x}$. We start by showing that $\mathrm{bf}^{\rho}(x) \subseteq \mathrm{bf}^{\rho_{i+1}}(x)$. Let $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho}(x)$. Denote $\boldsymbol{f} = R(\bar{a})$.

- Suppose that $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x)$. By definition of $\mathrm{bf}^{\rho_i}(x)$, there are values $y \in \mathcal{N}$, $t \in \mathbb{N}$ and $u \in \mathbb{N}$ such that $\mathtt{chosen}_R(y, t, x, u, \bar{a}) \in M$ and $j = \mathit{glob}_M(y, t) < i \leq \mathit{glob}_M(x, u)$. Now, since $x \neq x_i$, we more specifically have $i < \mathit{glob}_M(x, u)$ and thus $i + 1 \leq \mathit{glob}_M(x, u)$. Therefore $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, as desired.
- Suppose that $\langle j, \boldsymbol{f} \rangle \in \delta^{i \to x}$. By definition of $\delta^{i \to x}$, this implies $j = i$ and $R(x, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. By Claim B.7 we then have $\langle j, \boldsymbol{f} \rangle = \langle i, R(\bar{a}) \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, as desired.

Secondly, we show that $\mathrm{bf}^{\rho_{i+1}}(x) \subseteq \mathrm{bf}^{\rho}(x)$. Let $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$. Denote $\boldsymbol{f} = R(\bar{a})$. By definition of $\mathrm{bf}^{\rho_{i+1}}(x)$, there are values $y \in \mathcal{N}$, $t \in \mathcal{N}$ and $u \in \mathcal{N}$ such that $\mathtt{chosen}_R(y, t, x, u, \bar{a}) \in M$ and $j = \mathit{glob}_M(y, t) < i + 1 \leq \mathit{glob}_M(x, u)$. So $j \leq i$. We have the following cases:

- Suppose that $j < i$. Thus $\mathit{glob}_M(y, t) < i$. This immediately gives $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x) \subseteq \mathrm{bf}^{\rho}(x)$, as desired.
- Suppose that $j = i$. By Claim B.8 we then have $R(x, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. This implies that $\langle j, \boldsymbol{f} \rangle = \langle i, R(\bar{a}) \rangle \in \delta^{i \to x} \subseteq \mathrm{bf}^{\rho}(x)$, as desired.

**Case $x = x_i$.** By definition, $\mathrm{bf}^{\rho}(x) = (\mathrm{bf}^{\rho_i}(x) \setminus m_i) \cup \delta^{i \to x}$. Some parts of the reasoning are similar as for the case $x \neq x_i$. Repitition is avoided by referring to shared subclaims where possible.

We start by showing that $\mathrm{bf}^{\rho}(x) \subseteq \mathrm{bf}^{\rho_{i+1}}(x)$. Let $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho}(x)$. Denote $\boldsymbol{f} = R(\bar{a})$. We have the following cases:

- Suppose that $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x) \setminus m_i$. Thus $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x)$ and $\langle j, \boldsymbol{f} \rangle \notin m_i$. The former implies that there are values $y \in \mathcal{N}$, $t \in \mathbb{N}$ and $u \in \mathbb{N}$ such that $\mathtt{chosen}_R(y, t, x, u, \bar{a}) \in M$ and $j = \mathit{glob}_M(y, t) < i \leq \mathit{glob}_M(x, u)$. It must be that $\mathit{glob}_M(x, u) \neq i$, because otherwise $\langle j, \boldsymbol{f} \rangle \in m_i$, which is false. Combined with $i \leq \mathit{glob}_M(x, u)$, this gives $i < \mathit{glob}_M(x, u)$. Hence, $i + 1 \leq \mathit{glob}_M(x, u)$ and we obtain that $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, as desired.
- Suppose that $\langle j, \boldsymbol{f} \rangle \in \delta^{i \to x}$. By definition of $\delta^{i \to x}$, this implies $j = i$ and $R(x, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. By Claim B.7 we then have $\langle i, R(\bar{a}) \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, as desired.

Secondly, we show that $\mathrm{bf}^{\rho_{i+1}}(x) \subseteq \mathrm{bf}^{\rho}(x)$. Let $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$. Denote $\boldsymbol{f} = R(\bar{a})$. By definition of $\mathrm{bf}^{\rho_{i+1}}(x)$, there are values $y \in \mathcal{N}$, $t \in \mathbb{N}$ and $u \in \mathbb{N}$ such that $\mathtt{chosen}_R(y, t, x, u, \bar{a}) \in M$ and $j = \mathit{glob}_M(y, t) < i + 1 \leq \mathit{glob}_M(x, u)$. Now we look at the cases for $j$:

- Suppose that $j < i$. This gives us $\mathit{glob}_M(y, t) < i \leq \mathit{glob}_M(x, u)$, which implies $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x)$. Moreover, $i + 1 \leq \mathit{glob}_M(x, u)$ implies that $\mathit{glob}_M(x, u) \neq i$. Hence, $\langle j, \boldsymbol{f} \rangle \notin m_i$. Taken together, we now have $\langle j, \boldsymbol{f} \rangle \in \mathrm{bf}^{\rho_i}(x) \setminus m_i \subseteq \mathrm{bf}^{\rho}(x)$.
- Suppose that $j = i$. Then $\langle i, R(\bar{a}) \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, and by Claim B.8 we obtain that $R(x, \bar{a}) \in \mathit{async}_{\mathcal{P}}\langle D_i \rangle$. Therefore $\langle j, \boldsymbol{f} \rangle = \langle i, R(\bar{a}) \rangle \in \delta^{i \to x} \subseteq \mathrm{bf}^{\rho}(x)$, as desired.

## B.4 Trace

In this section we show that $\mathit{trace}(\mathcal{R}) = M|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}}$. Recall the definition of $\mathit{trace}(\mathcal{R})$ from Section 4.2:

$$\mathit{trace}(\mathcal{R}) = \bigcup_{i \in \mathbb{N}} (D_i)^{\Uparrow x_i,\, \mathit{loc}_{\mathcal{R}}(i)}.$$

For each $i \in \mathbb{N}$, we have defined $loc_{\mathcal{R}}(i)$ as the number of transitions in $\mathcal{R}$ that come before $i$ and in which $x_i$ is also the active node. It follows from the construction of $\mathcal{R}$ that $loc_{\mathcal{R}}(i) = loc_M(i, x_i)$. Indeed, $loc_M(i, x_i)$ counts the number of pairs in $\mathcal{N} \times \mathbb{N}$ with node $x_i$ that have an ordinal strictly smaller than $i$, and that is precisely the number of transitions in $\mathcal{R}$ with active node $x_i$ that come before $i$. Moreover, by Claim B.2 we have $loc_M(i, x_i) = s_i$. Hence,

$$trace(\mathcal{R}) = \bigcup_{i \in \mathbb{N}} (D_i)^{\Uparrow x_i, s_i}.$$

By Claim B.9, for each $i \in \mathbb{N}$ we have $(D_i)^{\Uparrow x_i, s_i} = M|^{x_i, s_i}$, and thus

$$trace(\mathcal{R}) = \bigcup_{i \in \mathbb{N}} M|^{x_i, s_i}.$$

For the next step, let us denote $A = \{(x_i, s_i) \mid i \in \mathbb{N}\}$. We show that $A = \mathcal{N} \times \mathbb{N}$. First, it follows from their respective definitions that $x_i \in \mathcal{N}$ and $s_i \in \mathbb{N}$, and therefore $A \subseteq \mathcal{N} \times \mathbb{N}$. Secondly, we show that $\mathcal{N} \times \mathbb{N} \subseteq A$. Let $(x, s) \in \mathcal{N} \times \mathbb{N}$. Denote $i = glob_M(x, s)$. By definition, $x_i = x$ and $s_i = s$. Hence $(x, s) = (x_i, s_i) \in A$. Now we may write:

$$\begin{aligned} trace(\mathcal{R}) &= \bigcup_{(x,s) \in A} M|^{x,s} \\ &= \bigcup_{(x,s) \in \mathcal{N} \times \mathbb{N}} M|^{x,s}. \end{aligned}$$

Finally, because $M$ is well-formed (see Section 7), there are no facts in $M|_{sch(\mathcal{P})^{\text{LT}}}$ of the form $R(v, w, \bar{a})$ with $v \notin \mathcal{N}$ or $w \notin \mathbb{N}$. We obtain, as desired:

$$trace(\mathcal{R}) = M|_{sch(\mathcal{P})^{\text{LT}}}.$$

## B.5 Subclaims

**Claim B.1.** Let $x \in \mathcal{N}$ and $s \in \mathbb{N}$. We have $(M|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s} = H(x)^{\Uparrow x,s}$.

*Proof.* First, by construction of $decl(H)$ we have $(decl(H)|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s} = H(x)^{\Uparrow x,s}$. Because $decl(H) \subseteq M$, and because facts over $edb(\mathcal{P})^{\text{LT}}$ can not be derived by rules in $pure(\mathcal{P})$, we have $M|_{edb(\mathcal{P})^{\text{LT}}} = decl(H)|_{edb(\mathcal{P})^{\text{LT}}}$. Hence,

$$(M|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s} = (decl(H)|_{edb(\mathcal{P})^{\text{LT}}})|^{x,s} = H(x)^{\Uparrow x,s},$$

and the result is obtained. □

---

**Claim B.2.** Let $i \in \mathbb{N}$. Recall the definition of $x_i$ and $s_i$ as the node and timestamp in the pair of $\mathcal{N} \times \mathbb{N}$ that has ordinal $i$ in $<_M$. We have $s_i = loc_M(i, x_i)$.

*Proof.* By definition of $x_i$ and $s_i$, we have $glob_M(x_i, s_i) = i$. Suppose we would know for all timestamp values $s \in \mathbb{N}$ and $t \in \mathbb{N}$ that $s < t$ implies $glob_M(x_i, s) < glob_M(x_i, t)$. Then $loc_M(i, x_i)$, which is the number of pairs from $\mathcal{N} \times \mathbb{N}$ with first component $x_i$ that come strictly before ordinal $i$, is precisely

$$|\{s \in \mathbb{N} \mid s < s_i\}|.$$

This equals $s_i$, as desired.

We are left to show for any $s \in \mathbb{N}$ and $t \in \mathbb{N}$ that $s < t$ implies $glob_M(x_i, s) < glob_M(x_i, t)$. It is actually sufficient to show for any $s \in \mathbb{N}$ that $(x_i, s) \prec_M (x_i, s+1)$. Indeed, this would imply for any $t \in \mathbb{N}$ with $s < t$ that

$$(x_i, s) \prec_M (x_i, s+1) \prec_M (x_i, s+2) \prec_M \ldots \prec_M (x_i, t).$$

And since $\prec_M$ is a partial order, it is transitive, and thus $(x_i, s) \prec_M (x_i, t)$. Next, since $<_M$ respects $\prec_M$, we obtain $(x_i, s) <_M (x_i, t)$ and thus $glob_M(x_i, s) < glob_M(x_i, t)$, as desired. To show that $(x_i, s) \prec_M (x_i, s+1)$, we observe that the rule (4.1) in $pure(\mathcal{P})$ is positive. Hence, for any $s \in \mathbb{N}$, the following ground rule is always in $gr_M^{\mathcal{P},H}$, and it derives $\texttt{before}(x_i, s, x_i, s+1) \in M$ because $\{\texttt{all}(x_i), \texttt{tsucc}(s, s+1)\} \subseteq decl(H) \subseteq M$:

$$\texttt{before}(x_i, s, x_i, s+1) \leftarrow \texttt{all}(x_i), \texttt{tsucc}(s, s+1).$$

Thus for all $s \in \mathbb{N}$ we have $(x_i, s) \prec_M (x_i, s+1)$ by definition of $\prec_M$. □

---

**Claim B.3.** Let $i \in \mathbb{N}$. We have $M^{\text{ind}}|^{x_i, s_i+1} \subseteq induc_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x_i, s_i+1}$.

*Proof.* Let $\boldsymbol{f} \in M^{\text{ind}}|^{x_i, s_i+1}$. We show that $\boldsymbol{f} \in induc_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x_i, s_i+1}$.

By definition of $M^{\text{ind}}$, there is an active *inductive* ground rule $\psi \in gr_M^{\mathcal{P},H}$ that can derive $\boldsymbol{f}$, i.e., $head_\psi = \boldsymbol{f}$. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in pure(\mathcal{P})$ and a valuation $V$ so that $\psi$ can be obtained from $\varphi$ by applying $V$ and by subsequently removing all negative (ground) body literals, and so that $V(neg_\varphi) \cap M = \emptyset$. The rule $\varphi$ must be of the form (4.4), which implies that $V$ must assign $x_i$ and $s_i$ to the body location and timestamp variable respectively, and that it must assign $x_i$ and $s_i+1$ to the head location and timestamp variable respectively.

Let $\varphi' \in \mathcal{P}$ be the original inductive rule on which $\varphi$ is based. Let $\varphi'' \in induc_{\mathcal{P}}$ be the rule corresponding to $\varphi'$. It follows from the construction of $\varphi$ out of $\varphi'$ and $\varphi''$ out of $\varphi'$ that valuation $V$ can also be applied to rule $\varphi''$. Indeed, rule $\varphi$ just has more variables for the location and timestamps. We show that $V$ is satisfying for $\varphi''$ with respect to $D_i$, so that $\varphi''$ and $V$ together derive $V(head_{\varphi''}) = \boldsymbol{f}^{\Downarrow} \in induc_{\mathcal{P}}\langle D_i \rangle$, which gives $\boldsymbol{f} \in induc_{\mathcal{P}}\langle D_i \rangle^{\Uparrow x_i, s_i+1}$, as desired.

We must concretely show that $V(pos_{\varphi''}) \subseteq D_i$ and $V(neg_{\varphi''}) \cap D_i = \emptyset$. We start by showing that $V(pos_{\varphi''}) \subseteq D_i$. From the relationship between $\psi$, $\varphi$ and $\varphi''$, we know that

$$pos_\psi|_{sch(\mathcal{P})^{\text{LT}}} = V(pos_\varphi)|_{sch(\mathcal{P})^{\text{LT}}} = V(pos_{\varphi''})^{\Uparrow x_i, s_i}.$$

Since $\psi$ is active with respect to $M$, we have $pos_\psi \subseteq M$, and thus $V(pos_{\varphi''})^{\Uparrow x_i,s_i} \subseteq M$. Then by Claim B.4 we have $V(pos_{\varphi''}) \subseteq D_i$, as desired.

Now we show that $V(neg_{\varphi''}) \cap D_i = \emptyset$. By the relationship of $\varphi$ and $\varphi''$, we have $V(neg_{\varphi''})^{\Uparrow x_i,s_i} \subseteq V(neg_\varphi)$. By choice of $\varphi$ and $V$, we have $V(neg_\varphi) \cap M = \emptyset$. Hence, $V(neg_{\varphi''})^{\Uparrow x_i,s_i} \cap M = \emptyset$. Finally, by Claim B.5, we have $V(neg_{\varphi''}) \cap D_i = \emptyset$, as desired. □

---

**Claim B.4.** Let $i \in \mathbb{N}$. Let $I$ be a set of facts over the schema $sch(\mathcal{P})^{\text{LT}}$ that all have location specifier $x_i$ and timestamp $s_i$. If $I \subseteq M$ then $I^{\Downarrow} \subseteq D_i$.

*Proof.* By the given assumptions on $I$, we have $I \subseteq M|^{x_i,s_i}$. Then by Claim B.9 we have $I \subseteq (D_i)^{\Uparrow x_i,s_i}$. Hence $I^{\Downarrow} \subseteq D_i$, as desired. □

---

**Claim B.5.** Let $i \in \mathbb{N}$. Let $I$ be a set of facts over the schema $sch(\mathcal{P})^{\text{LT}}$ that all have location specifier $x_i$ and timestamp $s_i$. If $I \cap M = \emptyset$ then $I^{\Downarrow} \cap D_i = \emptyset$.

*Proof.* We are given that $I \cap M = \emptyset$. This implies $I \cap M|^{x_i,s_i} = \emptyset$. By Claim B.9 we have $I \cap (D_i)^{\Uparrow x_i,s_i} = \emptyset$. Hence $I^{\Downarrow} \cap D_i = \emptyset$, as desired. □

---

**Claim B.6.** Let $i \in \mathbb{N}$. We have $induc_{\mathcal{P}}\langle D_i\rangle^{\Uparrow x_i,s_i+1} \subseteq M^{\text{ind}}|^{x_i,s_i+1}$.

*Proof.* Let $\boldsymbol{f} \in induc_{\mathcal{P}}\langle D_i\rangle$. We show that $\boldsymbol{f}^{\Uparrow x_i,s_i+1} \in M^{\text{ind}}|^{x_i,s_i+1}$.

Recall the semantics for $induc_{\mathcal{P}}$ from Section 3.3. Let $\varphi \in induc_{\mathcal{P}}$ and $V$ be the rule and valuation that together derived $\boldsymbol{f} \in induc_{\mathcal{P}}\langle D_i\rangle$. Let $\varphi' \in \mathcal{P}$ be the original inductive rule on which $\varphi$ is based. Let $\varphi'' \in pure(\mathcal{P})$ be the inductive rule that in turn is based on $\varphi'$, which is of the form (4.4). Let $V''$ be the valuation for $\varphi''$ that is obtained by extending $V$ to assign $x_i$ and $s_i$ to respectively the location and timestamp variables in the body, and to assign $s_i + 1$ to the head timestamp variable. Let $\psi$ be the positive ground rule obtained from $\varphi''$ by applying the valuation $V''$, and by subsequently removing the negative (ground) body literals. Note that $head_\psi = V(head_\varphi)^{\Uparrow x_i,s_i+1} = \boldsymbol{f}^{\Uparrow x_i,s_i+1}$. We will show that $\psi \in gr_M^{\mathcal{P},H}$ and that $pos_\psi \subseteq M$, so that this ground rule derives $\boldsymbol{f}^{\Uparrow x_i,s_i+1} \in M$. And since $\psi$ is inductive, we more specifically have $\boldsymbol{f}^{\Uparrow x_i,s_i+1} \in M^{\text{ind}}|^{x_i,s_i+1}$, as desired.

- First we show that $\psi \in gr_M^{\mathcal{P},H}$. This requires that $V''(neg_{\varphi''}) \cap M = \emptyset$. From the construction of rule $\varphi''$, we have $V''(neg_{\varphi''}) = V(neg_\varphi)^{\Uparrow x_i,s_i}$. We will show that $V(neg_\varphi)^{\Uparrow x_i,s_i} \cap M = \emptyset$.

  Because $V$ is satisfying for $\varphi$ with respect to $D_i$, we have $V(neg_\varphi) \cap D_i = \emptyset$. This gives $V(neg_\varphi)^{\Uparrow x_i,s_i} \cap (D_i)^{\Uparrow x_i,s_i} = \emptyset$. By Claim B.9 we then have $V(neg_\varphi)^{\Uparrow x_i,s_i} \cap M|^{x_i,s_i} = \emptyset$. As the set $V(neg_\varphi)^{\Uparrow x_i,s_i}$ contains only facts over $sch(\mathcal{P})^{\text{LT}}$ with location specifier $x_i$ and timestamp $s_i$, we have $V(neg_\varphi)^{\Uparrow x_i,s_i} \cap M = \emptyset$, as desired. We obtain that $\psi \in gr_M^{\mathcal{P},H}$.

- Now we show that $pos_\psi \subseteq M$. From the construction of rule $\varphi''$, we have

$$pos_\psi = V''(pos_{\varphi''}) = V(pos_\varphi)^{\Uparrow x_i, s_i} \cup \{\texttt{tsucc}(s_i, s_i + 1)\}.$$

  We immediately have $\texttt{tsucc}(s_i, s_i + 1) \in \mathit{decl}(H) \subseteq M$. Moreover, since $V$ is satisfying for $\varphi$ with respect to $D_i$, we have $V(pos_\varphi) \subseteq D_i$. Hence $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq (D_i)^{\Uparrow x_i, s_i}$. By Claim B.9 we then have $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq M|^{x_i, s_i} \subseteq M$, as desired.

□

---

**Claim B.7.** Let $i \in \mathbb{N}$. Let $x \in \mathcal{N}$. For each fact $R(x, \bar{a}) \in \mathit{async}_\mathcal{P}\langle D_i \rangle$, we have $\langle i, R(\bar{a}) \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$. Intuitively, this means that the messages that are sent under the operational semantics are also sent in $M$, since $\mathrm{bf}^{\rho_{i+1}}(x)$ is defined purely in terms of $M$.

*Proof.* The main approach of this proof is as follows. We will show that there is a timestamp $u \in \mathbb{N}$ such that $\texttt{chosen}_R(x_i, s_i, x, u, \bar{a}) \in M$. Next, because rules of the form (4.9) are positive, in $gr_M^{\mathcal{P},H}$ there is always the following ground rule:

$$\texttt{before}(x_i, s_i, x, u) \leftarrow \texttt{chosen}_R(x_i, s_i, x, u, \bar{a}).$$

Thus if $\texttt{chosen}_R(x_i, s_i, x, u, \bar{a}) \in M$ then $\texttt{before}(x_i, s_i, x, u) \in M$, which implies that $(x_i, s_i) \prec_M (x, u)$ by definition of $\prec_M$. Since $<_M$ respects $\prec_M$, we obtain $(x_i, s_i) <_M (x, u)$ and thus $\mathit{glob}_M(x_i, s_i) < \mathit{glob}_M(x, u)$. Also, since $\mathit{glob}_M(x_i, s_i) = i$, we overall get

$$\mathit{glob}_M(x_i, s_i) < i + 1 \leq \mathit{glob}_M(x, u),$$

which together with $\texttt{chosen}_R(x_i, s_i, x, u, \bar{a}) \in M$ gives $\langle \mathit{glob}_M(x_i, s_i), R(\bar{a}) \rangle = \langle i, R(\bar{a}) \rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, as desired.

Now we are left to show that such a timestamp $u$ exists. Recall the semantics for $\mathit{async}_\mathcal{P}$ from Section 3.3. Let $\varphi \in \mathit{async}_\mathcal{P}$ and $V$ be a rule and valuation that together derived $R(x, \bar{a}) \in \mathit{async}_\mathcal{P}\langle D_i \rangle$. Let $\varphi' \in \mathcal{P}$ be the original asynchronous rule on which $\varphi$ is based. Let $\varphi'' \in \mathit{pure}(\mathcal{P})$ be the rule obtained by applying transformation (4.5) to $\varphi'$. To continue, because $\prec_M$ is well-founded, there are only a finite number of timestamps $v \in \mathbb{N}$ of node $x$ such that $(x, v) \prec_M (x_i, s_i)$. So, there exists a timestamp $u \in \mathbb{N}$ such that $(x, u) \not\prec_M (x_i, s_i)$. Now, let $V''$ be the valuation for $\varphi''$ that is the extension of valuation $V$ to assign $x_i$ and $s_i$ to the body location variable and timestamp variable respectively (both belonging to the sender), and to assign $u$ to the addressee arrival timestamp. Note that from the construction of $\varphi''$ we also know that $V$ (and thus $V''$) assigns the value $x$ to the addressee location variable and the tuple $\bar{a}$ to the message contents. Let $\psi$ denote the ground rule obtained by applying $V''$ to $\varphi''$, and by subsequently removing the negative (ground) body literals. We will first show that $\psi \in gr_M^{\mathcal{P},H}$, and then we show that $pos_\psi \subseteq M$, meaning

that $\psi$ derives $head_\psi = \texttt{cand}_R(x_i, s_i, x, u, \bar{a}) \in M$. Then Claim B.11 can be applied to know that there is a timestamp $u'$, with possibly $u' = u$, such that $\texttt{chosen}_R(x_i, s_i, x, u', \bar{a}) \in M$, as desired.

In order for $\psi$ to be in $gr_M^{\mathcal{P},H}$, it is required that $V''(neg_{\varphi''}) \cap M = \emptyset$. It follows from the construction of $\varphi''$ out of $\varphi'$ and $\varphi$ out of $\varphi'$ that

$$V''(neg_{\varphi''}) = V(neg_\varphi)^{\Uparrow x_i, s_i} \cup \{\texttt{before}(x, u, x_i, s_i)\}.$$

It must be that $\texttt{before}(x, u, x_i, s_i) \notin M$ because otherwise $(x, u) \prec_M (x_i, s_i)$ by definition of $\prec_M$, which is false by choice of $u$. Next, we show that $V(neg_\varphi)^{\Uparrow x_i, s_i} \cap M = \emptyset$. Because $V$ is satisfying for $\varphi$ with respect to $D_i$, we have $V(neg_\varphi) \cap D_i = \emptyset$, and thus

$$V(neg_\varphi)^{\Uparrow x_i, s_i} \cap (D_i)^{\Uparrow x_i, s_i} = \emptyset.$$

Moreover, by Claim B.9 we have $(D_i)^{\Uparrow x_i, s_i} = M|^{x_i, s_i}$, and thus

$$V(neg_\varphi)^{\Uparrow x_i, s_i} \cap M|^{x_i, s_i} = \emptyset.$$

As the set $V(neg_\varphi)^{\Uparrow x_i, s_i}$ contains only facts over $sch(\mathcal{P})^{\mathrm{LT}}$ with location specifier $x_i$ and timestamp $s_i$, we have

$$V(neg_\varphi)^{\Uparrow x_i, s_i} \cap M = \emptyset,$$

as desired. We obtain that $\psi \in gr_M^{\mathcal{P},H}$.

For the last part, we will show that $\psi$ is active on $M$, meaning that $pos_\psi \subseteq M$. Note that $pos_\psi = V''(pos_{\varphi''})$. From the construction of $\varphi''$ we have

$$V''(pos_{\varphi''}) = V(pos_\varphi)^{\Uparrow x_i, s_i} \cup \{\texttt{all}(x), \texttt{time}(u)\}.$$

Because $x \in \mathcal{N}$ and $u \in \mathbb{N}$, we immediately have $\{\texttt{all}(x), \texttt{time}(u)\} \subseteq decl(H) \subseteq M$. We are left to show that $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq M$. Because $V$ is satisfying for $\varphi$ with respect to $D_i$, we have $V(pos_\varphi) \subseteq D_i$. Hence $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq (D_i)^{\Uparrow x_i, s_i}$. By again using Claim B.9 we then obtain $V(pos_\varphi)^{\Uparrow x_i, s_i} \subseteq M|^{x_i, s_i} \subseteq M$, as desired. □

---

**Claim B.8.** Let $i \in \mathbb{N}$ and $x \in \mathcal{N}$. For each pair $\langle i, R(\bar{a})\rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$, we have $R(x, \bar{a}) \in async_{\mathcal{P}}\langle D_i\rangle$. Note the send-tag $i$ in the pair $\langle i, R(\bar{a})\rangle$. Intuitively, this claim says that the messages that are sent in $M$ are also sent in the operational semantics.

*Proof.* By definition of $\mathrm{bf}^{\rho_{i+1}}(x)$, the pair $\langle i, R(\bar{a})\rangle \in \mathrm{bf}^{\rho_{i+1}}(x)$ implies that there are values $y \in \mathcal{N}$, $t \in \mathbb{N}$ and $u \in \mathbb{N}$ such that $\texttt{chosen}_R(y, t, x, u, \bar{a}) \in M$, $glob_M(y, t) = i$ and $glob_M(y, t) < i + 1 \leq glob_M(x, u)$. And $glob_M(y, t) = i$ gives us that $y = x_i$ and $t = s_i$. Thus $\texttt{chosen}_R(x_i, s_i, x, u, \bar{a}) \in M$.

All ground rules in $gr_M^{\mathcal{P},H}$ that can derive $\texttt{chosen}_R(x_i, s_i, x, u, \bar{a}) \in M$ are of the form (4.6), and hence $\texttt{cand}_R(x_i, s_i, x, u, \bar{a}) \in M$. Let $\psi \in gr_M^{\mathcal{P},H}$ be

an active ground rule with head $\texttt{cand}_R(x_i, s_i, x, u, \bar{a})$. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in \mathit{pure}(\mathcal{P})$ and a valuation $V$ so that $\psi$ is obtained from $\varphi$ by applying $V$ and by subsequently removing all negative (ground) body literals, and so that $V(\mathit{neg}_\varphi) \cap M = \emptyset$. The rule $\varphi$ is of the form (4.5), which implies that $V$ must assign $x_i$ and $s_i$ respectively to the body location and timestamp variable that correspond to the sender, and that it must assign $x$ and $u$ respectively to the location and timestamp variable that correspond to the addressee. Let $\varphi' \in \mathcal{P}$ be the original asynchronous rule on which $\varphi$ is based. Let $\varphi''$ be the corresponding rule in $\mathit{async}_\mathcal{P}$. From the construction of $\varphi$ out of $\varphi'$ and $\varphi''$ out of $\varphi'$, it follows that $V$ can also be applied to $\varphi''$. Note that $V(\mathit{head}_{\varphi''}) = R(x, \bar{a})$. We now show that $V$ is satisfying for $\varphi''$ with respect to $D_i$, which causes $R(x, \bar{a}) \in \mathit{async}_\mathcal{P}\langle D_i \rangle$, as desired. Specifically, we have to show that $V(\mathit{pos}_{\varphi''}) \subseteq D_i$ and $V(\mathit{neg}_{\varphi''}) \cap D_i = \emptyset$.

First we show that $V(\mathit{pos}_{\varphi''}) \subseteq D_i$. By construction of $\varphi$ and $\varphi''$, we have

$$\mathit{pos}_\psi|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}} = V(\mathit{pos}_\varphi)|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}} = V(\mathit{pos}_{\varphi''})^{\Uparrow x_i, s_i}.$$

Since $\psi$ is active, we have $\mathit{pos}_\psi|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}} \subseteq M$, and therefore $V(\mathit{pos}_{\varphi''})^{\Uparrow x_i, s_i} \subseteq M$. Then, because the facts in $V(\mathit{pos}_{\varphi''})^{\Uparrow x_i, s_i}$ are over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$ and have location specifier $x_i$ and timestamp $s_i$, we can apply Claim B.4 to know that $V(\mathit{pos}_{\varphi''}) \subseteq D_i$, as desired.

Now we show that $V(\mathit{neg}_{\varphi''}) \cap D_i = \emptyset$. By construction of $\varphi$ and $\varphi''$, we have

$$V(\mathit{neg}_\varphi)|_{\mathit{sch}(\mathcal{P})^{\mathrm{LT}}} = V(\mathit{neg}_{\varphi''})^{\Uparrow x_i, s_i}.$$

By choice of $\varphi$ and $V$, we have $V(\mathit{neg}_\varphi) \cap M = \emptyset$. Hence, $V(\mathit{neg}_{\varphi''})^{\Uparrow x_i, s_i} \cap M = \emptyset$. Then, because the facts in $V(\mathit{neg}_{\varphi''})^{\Uparrow x_i, s_i}$ are over $\mathit{sch}(\mathcal{P})^{\mathrm{LT}}$ and have location specifier $x_i$ and timestamp $s_i$, we can apply Claim B.5 to know that $V(\mathit{neg}_{\varphi''}) \cap D_i = \emptyset$, as desired. □

---

**Claim B.9.** Let $i \in \mathbb{N}$. We have $M|^{x_i, s_i} = (D_i)^{\Uparrow x_i, s_i}$. Intuitively, this means that the operational deductive fixpoint during transition $i$, corresponding to step $s_i$ of node $x_i$, is represented by $M$ in an exact way.

*Proof.* Let $n$ denote the largest stratum number of the deductive rules of $\mathcal{P}$. We show by induction on $k = 0, 1, \ldots, n$ that

$$(M^{\to k})|^{x_i, s_i} = (D_i^{\to k})^{\Uparrow x_i, s_i}.$$

This will give us $(M^{\to n})|^{x_i, s_i} = (D_i^{\to n})^{\Uparrow x_i, s_i} = (D_i)^{\Uparrow x_i, s_i}$. Moreover, Claim B.12 says that $(M^{\to n})|^{x_i, s_i} = M|^{x_i, s_i}$, and thus we obtain $M|^{x_i, s_i} = (D_i)^{\Uparrow x_i, s_i}$, as desired.

**Base case ($k = 0$)** By definition,

$$M^{\to 0} = M^{\blacktriangle} \cup M^{\mathrm{duc},0}.$$

But since there are no deductive ground rules in $gr_M^{\mathcal{P},H}$ with stratum 0, we have $M^{\mathrm{duc},0} = \emptyset$. Hence,

$$\begin{aligned}(M^{\to 0})|^{x_i,s_i} &= (M^{\blacktriangle})|^{x_i,s_i} \\ &= (M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s_i} \cup M^{\mathrm{ind}}|^{x_i,s_i} \cup M^{\mathrm{deliv}}|^{x_i,s_i}. \qquad \text{(B.1)}\end{aligned}$$

In order to simplify expression (B.1), we make the following observations. First, by Claim B.10, we have

$$\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i,s_i} = (M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s_i} \cup M^{\mathrm{ind}}|^{x_i,s_i}.$$

Secondly, by Claim B.13 we have $M^{\mathrm{deliv}}|^{x_i,s_i} = \mathit{untag}(m_i)^{\Uparrow x_i,s_i}$. Now expression (B.1) can be rewritten to obtain the desired equality:

$$\begin{aligned}(M^{\to 0})|^{x_i,s_i} &= \mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i,s_i} \cup \mathit{untag}(m_i)^{\Uparrow x_i,s_i} \\ &= (\mathrm{st}^{\rho_i}(x_i) \cup \mathit{untag}(m_i))^{\Uparrow x_i,s_i} \\ &= (D_i^{\to 0})^{\Uparrow x_i,s_i}.\end{aligned}$$

**Induction hypothesis** For the induction hypothesis, we assume for a stratum number $k \geq 1$ that

$$(M^{\to k-1})|^{x_i,s_i} = (D_i^{\to k-1})^{\Uparrow x_i,s_i}.$$

**Inductive step** We show that

$$(M^{\to k})|^{x_i,s_i} = (D_i^{\to k})^{\Uparrow x_i,s_i}.$$

We show both inclusions separately, in Claims B.14 and B.15. □

---

**Claim B.10.** Let $i \in \mathbb{N}$. We have $\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i,s_i} = (M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s_i} \cup M^{\mathrm{ind}}|^{x_i,s_i}$.

*Proof.* By definition,

$$\mathrm{st}^{\rho_i}(x_i) = \left((M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s} \cup M^{\mathrm{ind}}|^{x_i,s}\right)^{\Downarrow},$$

where $s = \mathit{loc}_M(i, x_i)$. Using Claim B.2, we have $s = s_i$. Therefore,

$$\mathrm{st}^{\rho_i}(x_i)^{\Uparrow x_i,s_i} = (M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s_i} \cup M^{\mathrm{ind}}|^{x_i,s_i}.$$

□

---

**Claim B.11.** For each fact $\texttt{cand}_R(x, s, y, u, \bar{a}) \in M$, there is a timestamp $u' \in \mathbb{N}$ such that $\texttt{chosen}_R(x, s, y, u', \bar{a}) \in M$, with possibly $u' = u$. This intuitively means that if for a message there are candidate arrival times in $M$, then a real arrival time is effectively chosen.

*Proof.* We show this with a proof by contradiction. So, suppose that there is no such timestamp $u'$. Now, because $\texttt{cand}_R(x, s, y, u, \bar{a}) \in M$, the following ground rule, which is of the form (4.6), can not be in $gr_M^{\mathcal{P},H}$, because otherwise $\texttt{chosen}_R(x, s, y, u, \bar{a}) \in M$, which is assumed not to be possible:

$$\texttt{chosen}_R(x, s, y, u, \bar{a}) \leftarrow \texttt{cand}_R(x, s, y, u, \bar{a}).$$

Because rules of the form (4.6) contain a negative $\texttt{other}_{\ldots}$-atom in their body, the absence of the above ground rule from $gr_M^{\mathcal{P},H}$ implies that $\texttt{other}_R(x, s, y, u, \bar{a}) \in M$. This $\texttt{other}_R$-fact must be derived by a ground rule of the form (4.7):

$$\texttt{other}_R(x, s, y, u, \bar{a}) \leftarrow \texttt{cand}_R(x, s, y, u, \bar{a}),\ \texttt{chosen}_R(x, s, y, u', \bar{a}),\ u \neq u'.$$

But this implies that $\texttt{chosen}_R(x, s, y, u', \bar{a}) \in M$, which is a contradiction. □

---

**Claim B.12.** Let $i \in \mathbb{N}$. Let $n$ denote the largest stratum number of the deductive rules of $\mathcal{P}$. We have $(M^{\to n})|^{x_i,s_i} = M|^{x_i,s_i}$.

*Proof.* First, since $M^{\to n} \subseteq M$, we immediately have $(M^{\to n})|^{x_i,s_i} \subseteq M|^{x_i,s_i}$.

Now, let $\boldsymbol{f} \in M|^{x_i,s_i}$. We have to show that $\boldsymbol{f} \in (M^{\to n})|^{x_i,s_i}$. Since $\boldsymbol{f}$ has location specifier $x_i$ and timestamp $s_i$, it is sufficient to show that $\boldsymbol{f} \in M^{\to n}$. We have the following cases:

- Suppose that $\boldsymbol{f} \in M|_{edb(\mathcal{P})^{\text{LT}}}$. Then $\boldsymbol{f} \in M^{\blacktriangle} \subseteq M^{\to n}$.
- Suppose that $\boldsymbol{f} \in M|_{idb(\mathcal{P})^{\text{LT}}}$. This means that there is an active ground rule $\psi \in gr_M^{\mathcal{P},H}$ with $head_\psi = \boldsymbol{f}$. As seen in Section B.1, the rule $\psi$ can be of three types: deductive, inductive and delivery. The last two cases would respectively imply $\boldsymbol{f} \in M^{\text{ind}}$ and $\boldsymbol{f} \in M^{\text{deliv}}$, which gives us $\boldsymbol{f} \in M^{\blacktriangle} \subseteq M^{\to n}$. For the deductive case, the rule $\psi$ has a stratum number no larger than $n$, and hence $\boldsymbol{f} \in M^{\text{duc},n} \subseteq M^{\to n}$.

□

---

**Claim B.13.** Let $i \in \mathbb{N}$. We have $M^{\text{deliv}}|^{x_i,s_i} = untag(m_i)^{\Uparrow x_i,s_i}$.

*Proof.* Let $\boldsymbol{f} \in M^{\text{deliv}}|^{x_i,s_i}$. We show that $\boldsymbol{f} \in untag(m_i)^{\Uparrow x_i,s_i}$. Denote $\boldsymbol{f} = R(x_i, s_i, \bar{a})$. By definition of $M^{\text{deliv}}$, there is an active delivery rule $\psi \in gr_M^{\mathcal{P},H}$ that derives $\boldsymbol{f}$:

$$R(x_i, s_i, \bar{a}) \leftarrow \texttt{chosen}_R(y, t, x_i, s_i, \bar{a}).$$

Because this rule is active, we have $\texttt{chosen}_R(y, t, x_i, s_i, \bar{a}) \in M$. Now, by definition of $x_i$ and $s_i$, we have $glob_M(x_i, s_i) = i$. Hence, $\langle glob_M(y, t), R(\bar{a})\rangle \in m_i$ and thus $R(\bar{a}) \in untag(m_i)$. Finally, we obtain that $\boldsymbol{f} = R(x_i, s_i, \bar{a}) \in untag(m_i)^{\Uparrow x_i, s_i}$, as desired.

Let $\boldsymbol{f} \in untag(m_i)^{\Uparrow x_i, s_i}$. We show that $\boldsymbol{f} \in M^{\text{deliv}}|^{x_i, s_i}$. Denote $\boldsymbol{f} = R(x_i, s_i, \bar{a})$. We have $R(\bar{a}) \in untag(m_i)$. Thus, there is some tag $j \in \mathbb{N}$ such that $\langle j, R(\bar{a})\rangle \in m_i$. By definition of $m_i$, there are values $y \in \mathcal{N}$, $t \in \mathbb{N}$, $z \in \mathcal{N}$ and $u \in \mathbb{N}$ such that

$$\texttt{chosen}_R(y, t, z, u, \bar{a}) \in M,$$

where $glob_M(y, t) = j$ and $glob_M(z, u) = i$. This implies $z = x_i$ and $u = s_i$. Hence, $\texttt{chosen}_R(y, t, x_i, s_i, \bar{a}) \in M$. Now, the following ground rule $\psi$ is in $gr_M^{\mathcal{P},H}$ because (delivery) rules of the form (4.8) are always positive:

$$R(x_i, s_i, \bar{a}) \leftarrow \texttt{chosen}_R(y, t, x_i, s_i, \bar{a}).$$

This rule derives $\boldsymbol{f} = R(x_i, s_i, \bar{a}) \in M$ because its body-fact is in $M$. Moreover, $\boldsymbol{f}$ has location specifier $x_i$ and timestamp $s_i$, and $\psi$ is an active delivery rule. Thus we more specifically have $\boldsymbol{f} \in M^{\text{deliv}}|^{x_i, s_i}$, as desired. □

---

**Claim B.14.** Let $i \in \mathbb{N}$. Let $k$ be a stratum number (thus $k \geq 1$). Suppose that

$$(M^{\to k-1})|^{x_i, s_i} = (D_i^{\to k-1})^{\Uparrow x_i, s_i}.$$

We have

$$(M^{\to k})|^{x_i, s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i, s_i}.$$

*Proof.* We consider the fixpoint computation of $M$, i.e., $M = \bigcup_{l \in \mathbb{N}} M_l$ with $M_0 = decl(H)$ and $M_l = T(M_{l-1})$ for each $l \geq 1$, where $T$ is the immediate consequence operator of $gr_M^{\mathcal{P},H}$. By the semantics of operator $T$, we have $M_{l-1} \subseteq M_l$.

We show by induction on $l = 0, 1, 2, \ldots$, that

$$(M_l \cap M^{\to k})|^{x_i, s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i, s_i}.$$

This will imply that

$$\left(\left(\bigcup_{l \in \mathbb{N}} M_l\right) \cap M^{\to k}\right)|^{x_i, s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i, s_i}.$$

Hence, we obtain, as desired

$$(M \cap M^{\to k})|^{x_i, s_i} = (M^{\to k})|^{x_i, s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i, s_i}.$$

Before we start with the induction, recall from Section B.1 that

$$\begin{aligned} M^{\to k} &= M^{\blacktriangle} \cup M^{\text{duc},k} \\ &= M|_{edb(\mathcal{P})^{\text{LT}}} \cup M^{\text{ind}} \cup M^{\text{deliv}} \cup M^{\text{duc},k}. \end{aligned}$$

**Base case ($l = 0$)** We have $M_0 = \mathit{decl}(H)$. Thus $M_0$ contains no facts derived by deductive, inductive or delivery ground rules. Therefore,

$$M_0 \cap M^{\to k} \quad = \quad M|_{\mathit{edb}(\mathcal{P})^{\mathrm{LT}}}.$$

Hence,

$$\begin{array}{rcl}(M_0 \cap M^{\to k})|^{x_i,s_i} & \subseteq & (M^{\blacktriangle})|^{x_i,s_i} \\ & \subseteq & (M^{\to k-1})|^{x_i,s_i}.\end{array}$$

And by using the given equality $(M^{\to k-1})|^{x_i,s_i} = (D_i^{\to k-1})^{\Uparrow x_i,s_i}$, we obtain, as desired:

$$\begin{array}{rcl}(M_0 \cap M^{\to k})|^{x_i,s_i} & \subseteq & (D_i^{\to k-1})^{\Uparrow x_i,s_i} \\ & \subseteq & (D_i^{\to k})^{\Uparrow x_i,s_i}.\end{array}$$

**Induction hypothesis** We assume for some $l \geq 1$ that

$$(M_{l-1} \cap M^{\to k})|^{x_i,s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i,s_i}.$$

**Inductive step** We show that

$$(M_l \cap M^{\to k})|^{x_i,s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i,s_i}.$$

Let $\boldsymbol{f} \in (M_l \cap M^{\to k})|^{x_i,s_i}$. If $\boldsymbol{f} \in M_{l-1}$ then $\boldsymbol{f} \in (M_{l-1} \cap M^{\to k})|^{x_i,s_i}$ and the induction hypothesis can be immediately applied. Now suppose that $\boldsymbol{f} \in M_l \setminus M_{l-1}$. This means that there is a ground rule $\psi \in gr_M^{\mathcal{P},H}$ that derived $\boldsymbol{f}$ during the application of the immediate consequence operator when going from ordinal $l-1$ to ordinal $l$. We have $\mathit{pos}_\psi \subseteq M_{l-1}$. We have seen in Section B.1 that $\psi$ can be of three types: deductive, inductive or a delivery rule. If $\psi$ is an inductive rule or a delivery rule then

$$\begin{array}{rcl}\boldsymbol{f} & \in & M^{\mathrm{ind}}|^{x_i,s_i} \cup M^{\mathrm{deliv}}|^{x_i,s_i} \\ & \subseteq & (M^{\blacktriangle})|^{x_i,s_i} \\ & \subseteq & (M^{\to k-1})|^{x_i,s_i} \\ & = & (D_i^{\to k-1})^{\Uparrow x_i,s_i} \\ & \subseteq & (D_i^{\to k})^{\Uparrow x_i,s_i}.\end{array}$$

Now suppose that $\psi$ is a deductive rule. If $\psi$ has stratum less than or equal to $k-1$, then $\boldsymbol{f} \in (M^{\to k-1})|^{x_i,s_i}$. In that case, the given equality $(M^{\to k-1})|^{x_i,s_i} = (D_i^{\to k-1})^{\Uparrow x_i,s_i}$ can be immediately applied to obtain that $\boldsymbol{f} \in (D_i^{\to k-1})^{\Uparrow x_i,s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i,s_i}$, as desired. Now suppose that $\psi$ has stratum $k$. Because $\psi \in gr_M^{\mathcal{P},H}$, there is a rule $\varphi \in \mathit{pure}(\mathcal{P})$ and valuation $V$ so that $\psi$ is obtained from $\varphi$ by applying valuation $V$ and subsequently removing the negative (ground) body literals, and so that $V(\mathit{neg}_\varphi) \cap M = \emptyset$. Let $\varphi' \in \mathcal{P}$ be the original deductive rule on which $\varphi$ is based. Recall from Section 3.3 that the deductive rules of $\mathcal{P}$ are

added unmodified to $deduc_{\mathcal{P}}$. Thus $\varphi' \in deduc_{\mathcal{P}}$. By construction of $\varphi$ out of $\varphi'$, we see that valuation $V$ can also be applied to rule $\varphi'$. We now show that $V$ is satisfying for $\varphi'$ during the computation of $D_i$, and thus specifically during the computation of stratum $k$ because $\varphi$ (and $\varphi'$) has stratum $k$. Since $V(head_\varphi) = head_\psi = \boldsymbol{f}$, this results in the derivation of $V(head_{\varphi'}) = \boldsymbol{f}^{\Downarrow} \in D_i^{\to k}$ and thus $\boldsymbol{f} \in (D_i^{\to k})^{\Uparrow x_i, s_i}$, as desired. It is sufficient to show that $V(pos_{\varphi'}) \subseteq D_i^{\to k}$ and $V(neg_{\varphi'}) \cap D_i^{\to k-1} = \emptyset$ because in a syntactic stratification, a rule-body can use relations positively if the stratum of those relations is not higher than the stratum of the rule itself, and a rule-body can use relations negatively only if those relations have a stratum that is strictly lower than the stratum of the rule itself.

- We show that $V(pos_{\varphi'}) \subseteq D_i^{\to k}$. First, by the relationship between $\varphi$ and $\varphi'$, and because valuation $V$ assigns $x_i$ and $s_i$ to respectively the body location variable and body timestamp variable of $\varphi$, we have $pos_\psi = V(pos_\varphi) = V(pos_{\varphi'})^{\Uparrow x_i, s_i}$. By choice of $\psi$, we have $pos_\psi \subseteq M_{l-1}$. If we could more specifically show that $pos_\psi \subseteq M^{\to k}$, then $pos_\psi \subseteq (M_{l-1} \cap M^{\to k})|^{x_i, s_i}$. Then the induction hypothesis can be applied to obtain $pos_\psi = V(pos_{\varphi'})^{\Uparrow x_i, s_i} \subseteq (D_i^{\to k})^{\Uparrow x_i, s_i}$, which results in $V(pos_{\varphi'}) \subseteq D_i^{\to k}$, as desired.

  Now we show that $pos_\psi \subseteq M^{\to k}$. Let $\boldsymbol{g} \in pos_\psi$. If $\boldsymbol{g} \in M^{\blacktriangle}$ then we immediately have $\boldsymbol{g} \in M^{\to k}$. Now suppose that $\boldsymbol{g} \notin M^{\blacktriangle}$. Since $pos_\psi \subseteq M|^{x_i, s_i}$, we have $\boldsymbol{g} \in M|^{x_i, s_i} \setminus M^{\blacktriangle}$. Then Claim B.12 implies that there is an active deductive ground rule $\psi' \in gr_M^{\mathcal{P}, H}$ with $head_{\psi'} = \boldsymbol{g}$. But we are working with a syntactic stratification, and thus the stratum of $\psi'$ can not be higher than the stratum of $\psi$, which is $k$. Hence $\boldsymbol{g} \in M^{\mathrm{duc}, k} \subseteq M^{\to k}$.

- We show that $V(neg_{\varphi'}) \cap D_i^{\to k-1} = \emptyset$. By choice of $\varphi$ and $V$, we have $V(neg_\varphi) \cap M = \emptyset$. Since $(M^{\to k-1})|^{x_i, s_i} \subseteq M$, we have

  $$V(neg_\varphi) \cap (M^{\to k-1})|^{x_i, s_i} = \emptyset.$$

  By applying the given equality $(M^{\to k-1})|^{x_i, s_i} = (D_i^{\to k-1})^{\Uparrow x_i, s_i}$, we then have $V(neg_\varphi) \cap (D_i^{\to k-1})^{\Uparrow x_i, s_i} = \emptyset$. By the relationship between $\varphi$ and $\varphi'$, we have $V(neg_\varphi) = V(neg_{\varphi'})^{\Uparrow x_i, s_i}$. Thus $V(neg_{\varphi'}) \cap D_i^{\to k-1} = \emptyset$, as desired.

□

---

**Claim B.15.** Let $i \in \mathbb{N}$. Let $k$ be a stratum number (thus $k \geq 1$). Suppose that

$$(M^{\to k-1})|^{x_i, s_i} = (D_i^{\to k-1})^{\Uparrow x_i, s_i}.$$

We have

$$(D_i^{\to k})^{\Uparrow x_i, s_i} \subseteq (M^{\to k})|^{x_i, s_i}.$$

*Proof.* Recall that the semantics of stratum $k$ in $deduc_{\mathcal{P}}$ is that of semi-positive Datalog$^{\neg}$, with input $D_i^{\rightarrow k-1}$. So, we can consider $D_i^{\rightarrow k}$ to be a fixpoint, i.e., as the set $\bigcup_{l\in\mathbb{N}} A_l$ with $A_0 = D_i^{\rightarrow k-1}$ and $A_l = T(A_{l-1})$ for each $l \geq 1$, where $T$ is the immediate consequence operator of stratum $k$ in $deduc_{\mathcal{P}}$. We show by induction on $l = 0$, 1, 2, etc, that

$$(A_l)^{\Uparrow x_i,s_i} \subseteq (M^{\rightarrow k})|^{x_i,s_i}.$$

This then gives us the desired result.

**Base case ($l = 0$)** We have $A_0 = D_i^{\rightarrow k-1}$. By applying the given equality, we obtain

$$(A_0)^{\Uparrow x_i,s_i} = (D_i^{\rightarrow k-1})^{\Uparrow x_i,s_i} = (M^{\rightarrow k-1})|^{x_i,s_i} \subseteq (M^{\rightarrow k})|^{x_i,s_i}.$$

**Induction hypothesis** We assume for some $l \geq 1$ that

$$(A_{l-1})^{\Uparrow x_i,s_i} \subseteq (M^{\rightarrow k})|^{x_i,s_i}.$$

**Inductive step** Let $\boldsymbol{f} \in A_l$. We show that $\boldsymbol{f}^{\Uparrow x_i,s_i} \in (M^{\rightarrow k})|^{x_i,s_i}$. If $\boldsymbol{f} \in A_{l-1}$ then the induction hypothesis can be immediately applied to obtain the desired result. Now suppose that $\boldsymbol{f} \in A_l \setminus A_{l-1}$. Let $\varphi \in deduc_{\mathcal{P}}$ and $V$ be respectively a rule with stratum $k$ and a valuation that together derived $\boldsymbol{f} \in A_l$. Let $\varphi' \in pure(\mathcal{P})$ be the rule that is obtained from $\varphi$ by applying transformation (4.3). Let $V'$ be the valuation that is obtained by extending $V$ to assign $x_i$ and $s_i$ respectively to the body location and timestamp variable, which are also both used in the head (by construction of $\varphi'$). Let $\psi$ be the ground rule obtained from $\varphi'$ by applying valuation $V'$ and by subsequently removing all negative body literals. We show that $\psi \in gr_M^{\mathcal{P},H}$ and that $pos_\psi \subseteq M$. This then implies

$$head_\psi = V'(head_{\varphi'}) = V(head_\varphi)^{\Uparrow x_i,s_i} = \boldsymbol{f}^{\Uparrow x_i,s_i} \in M.$$

Moreover, because $\varphi$ (and thus $\varphi'$) has stratum $k$, rule $\psi$ is an active deductive ground rule with stratum $k$, and thus $\boldsymbol{f}^{\Uparrow x_i,s_i} \in (M^{\mathrm{duc},k})|^{x_i,s_i} \subseteq (M^{\rightarrow k})|^{x_i,s_i}$, as desired.

- We show that $\psi \in gr_M^{\mathcal{P},H}$. We have to show that $V'(neg_{\varphi'}) \cap M = \emptyset$. Because $V$ is satifying for $\varphi$, and because negation is only applied to lower strata, we have

  $$V(neg_\varphi) \cap D_i^{\rightarrow k-1} = \emptyset.$$

  Thus

  $$V(neg_\varphi)^{\Uparrow x_i,s_i} \cap (D_i^{\rightarrow k-1})^{\Uparrow x_i,s_i} = \emptyset.$$

  By the relationship between $\varphi$ and $\varphi'$, we have $V(neg_\varphi)^{\Uparrow x_i,s_i} = V'(neg_{\varphi'})$, which gives us

  $$V'(neg_{\varphi'}) \cap (D_i^{\rightarrow k-1})^{\Uparrow x_i,s_i} = \emptyset.$$

And by using the given equality $(M^{\to k-1})|^{x_i,s_i} = (D_i^{\to k-1})^{\Uparrow x_i,s_i}$, we have

$$V'(neg_{\varphi'}) \cap (M^{\to k-1})|^{x_i,s_i} = \emptyset.$$

Now, for the last step, we work towards a contradiction: suppose that there is a fact $\boldsymbol{g} \in V'(neg_{\varphi'}) \cap M$. From the construction of $\varphi'$, we know that $\boldsymbol{g}$ is over $sch(\mathcal{P})^{\mathrm{LT}}$ and has location specifier $x_i$ and timestamp $s_i$.

- If $\boldsymbol{g}$ is over $edb(\mathcal{P})^{\mathrm{LT}}$ then $\boldsymbol{g} \in (M|_{edb(\mathcal{P})^{\mathrm{LT}}})|^{x_i,s_i}$. Thus $\boldsymbol{g} \in (M^{\blacktriangle})|^{x_i,s_i} \subseteq (M^{\to k-1})|^{x_i,s_i}$, which is a contradiction.
- If $\boldsymbol{g}$ is over $idb(\mathcal{P})^{\mathrm{LT}}$ then there is an active ground rule $\psi' \in gr_M^{\mathcal{P},H}$ with $head_{\psi'} = \boldsymbol{g}$. As seen in Section B.1, the rule $\psi'$ is either deductive, inductive or a delivery rule. The last two cases would imply that $\boldsymbol{g} \in (M^{\mathrm{ind}} \cup M^{\mathrm{deliv}})|^{x_i,s_i} \subseteq (M^{\blacktriangle})|^{x_i,s_i}$, which gives a contradiction like in the previous case. Now suppose that $\psi'$ is deductive. Because the predicate of $\boldsymbol{g}$ is used negatively in $\varphi'$ and thus negatively in $\varphi$, the syntactic stratification assigns a smaller stratum number to $\psi'$ than the stratum number of $\psi$, which is $k$. Hence, $\boldsymbol{g} \in (M^{\to k-1})|^{x_i,s_i}$, which is again a contradiction.

We conclude that $V'(neg_{\varphi'}) \cap M = \emptyset$.

- We show that $pos_\psi \subseteq M$. Because $V$ is satisfying for $\varphi$, we have

$$V(pos_\varphi) \subseteq A_{l-1}.$$

By the relationship between $\varphi$ and $\varphi'$ (and $\psi$), we have $V(pos_\varphi)^{\Uparrow x_i,s_i} = V'(pos_{\varphi'}) = pos_\psi$. Thus

$$pos_\psi \subseteq (A_{l-1})^{\Uparrow x_i,s_i}.$$

By now applying the induction hypothesis, we obtain, as desired:

$$pos_\psi \subseteq (M^{\to k})|^{x_i,s_i} \subseteq M.$$

□

---