A Declarative Semantics for Dedalus

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

Electrical Engineering and Computer Sciences
University of California at Berkeley

Technical Report No. UCB/EECS-2011-120
http://www.eecs.berkeley.edu/Pubs/TechRpts/2011/EECS-2011-120.html

November 29, 2011

A Declarative Semantics for Dedalus

Peter Alvaro
UC Berkeley
palvaro@cs.berkeley.edu

Tom J. Ameloot*
Hasselt University & Transnational University of Limburg
tom.ameloot@uhasselt.be

Joseph M. Hellerstein
UC Berkeley
hellerstein@cs.berkeley.edu

William R. Marczak
UC Berkeley
wrm@cs.berkeley.edu

Jan Van den Bussche
Hasselt University & Transnational University of Limburg
jan.vandenbussche@uhasselt.be

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 original Dedalus rules, modified to account for nondeterministic choices, and augmented with control rules which model causality. The main result then is that, as far as fair 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 [24]. One of the latest incarnations of this approach is Dedalus [7, 8, 20], a Datalog-inspired language that has influenced other recent language designs for distributed and cloud computing such as WebDamLog [3] and Bloom [6]. Indeed, issues related to Dedalus and data-oriented distributed computing are recently receiving some attention at database theory conferences [19, 9, 3, 10, 29].

*PhD Fellow of the Fund for Scientific Research, Flanders (FWO).

It is well understood how an *operational* semantics for Dedalus-like languages can be defined formally [15, 25, 18, 9, 3, 26]. 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. In addition, the transition system is highly nondeterministic, because nodes work concurrently, communication is asynchronous, and messages can be delayed and eventually delivered out of order by the network.

A purely *declarative* formal semantics for the languages used in declarative (!) networking, however, has been lacking so far. 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 P on an input distributed database instance I. Runs represent distributed computations and, due to the nondeterminism mentioned above, there are typically many fair runs of P on I.

2. Each run can be completely 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 also includes information about the vector clocks associated to the run. Vector clocks are the standard mechanism to extract the "causal order" from a distributed computation [12]; this causal order (which is a partial order) relates the local steps of the different compute nodes through chains of communicated messages.

3. The main idea now is that the set of traces of runs can be obtained precisely as the set of *stable models* [17] of P on I. A few manipulations are needed before we can aim for such a result, however, because the Dedalus program P is not really a Datalog$^\neg$ program, due to the "asynchronous rules" that are used for sending messages. First, these rules, which involve a choice construct, are transformed into Datalog$^\neg$ rules

that simulate nondeterministic choice [22, 27]. Furthermore, P is augmented with a fixed, instance- and network-independent set of rules that express the vector clock mechanism. Finally, a simple definition for the notion of "fair" model is proposed. Our main result can then be proved, to the effect that the traces of fair runs are exactly the fair stable models of the obtained Datalog$^\neg$ program.

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. It is also interesting in its own right to see a rigorous proof that vector clocks can be expressed in Datalog$^\neg$ under the stable model semantics, capturing the operational notion of causality in a model-theoretic fashion. On a more technical note, we expand upon prior work on non-deterministic choice in logic programs: the correct expression of choice rules by Datalog$^\neg$ rules under the stable model semantics has so far only been proven rigorously when the original choice rules are part of a Datalog program without negation [27]. In our work, the semantics is proven correct in the presence of negation; as a matter of fact, the vector clock rules are not even locally stratified.

Perhaps most importantly, the result is of interest for grounding a representative database language for distributed and cloud computating in a well-motivated model-theoretic semantics. Indeed our characterization provides a purely declarative axiomatization of fair distributed program behaviors in terms of the stable models of a logical theory (finite set of Datalog$^\neg$ rules).

To conclude this introduction we mention that our paper fits in the more general theme of data management in networked and peer-to-peer settings, which has been important for some time, e.g., [1, 2, 13]. In recent years we are also seeing a more general resurgence of interest in Datalog, e.g., [14, 21].

This paper is organized as follows. In Section 2 we give preliminaries, including the language Dedalus. In Sections 3 and 4 we present the operational and declarative semantics for Dedalus. In Section 5 we define the trace for each run and in Section 6 we give our main result that relates the operational and declarative semantics. We conclude in Section 7.

2. PRELIMINARIES

A *database schema* $\mathcal{D}$ is a finite 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 also write (R, k) as $R^{(k)}$. To sometimes improve readability, as a slight abuse of notation, we treat $\mathcal{D}$ as a set of only relation names (without associated arities).

We assume a countably infinite universe **dom** of atomic data values that includes the set $\mathbb{N}$ of natural numbers. 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 set of facts in I whose predicate occurs as a relation name in $\mathcal{D}'$.

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

Location specifier and timestamp.

Let $\mathcal{D}$ be a database schema. We want to associate a *location specifier* and a discrete *timestamp* to facts over $\mathcal{D}$. The intuition of a fact $R(x, s, \bar{v})$ with location specifier x and timestamp s will be that $R(\bar{v})$ holds in node x at step s, where s is the *local* time in x. We will always take timestamps to be values in $\mathbb{N}$. Formally, we do this by considering facts over extended schemas of $\mathcal{D}$, as follows.

First, we write $\mathcal{D}^{\mathrm{L}}$ to denote the database schema obtained from $\mathcal{D}$ by incrementing the arity of every relation by one. The extra component for every relation will contain the location specifier, which is by convention the first component of a fact. For a database instance I over $\mathcal{D}^{\mathrm{L}}$ we write $I|^x$ to denote the facts of I that have location specifier x.

Secondly, we write $\mathcal{D}^{\mathrm{LT}}$ to denote the database schema obtained from $\mathcal{D}$ 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. For a database instance I over $\mathcal{D}^{\mathrm{LT}}$ we write $I|^{x,s}$ to denote the facts of I that have location specifier x and timestamp s.

For a fact $\boldsymbol{f}$ over $\mathcal{D}^{\mathrm{L}}$ and a timestamp $s \in \mathbb{N}$, we write $\boldsymbol{f}^{\uparrow s}$ to denote the fact over $\mathcal{D}^{\mathrm{LT}}$ obtained by adding timestamp s to $\boldsymbol{f}$. For example, $R(x, a)^{\uparrow 5} = R(x, 5, a)$. We also use this notation for a set of facts, with the meaning that the timestamp is added to every fact of the set. Conversely, for a fact $\boldsymbol{f}$ over schema $\mathcal{D}^{\mathrm{LT}}$ we write $\boldsymbol{f}^{\downarrow}$ to denote the fact over schema $\mathcal{D}^{\mathrm{L}}$ obtained by removing the timestamp. For example, $R(x, 5, a)^{\downarrow} = R(x, a)$.

2.1 Network and distributed databases

A *network* is a nonempty finite set of *nodes* $\mathcal{N}$, which are values in **dom**. Intuitively, $\mathcal{N}$ represents a set of identifiers of compute nodes involved in a distributed system. Explicit edges (channels) are not necessary because we work in a model where any node x can send a message to any other node y, as long as x knows about y, from input tables or from messages that x 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 [24]. In that case we would restrict attention to programs where nodes only send messages to nodes to which they are linked; these nodes would again be provided as input.

A *distributed database instance* H over $\mathcal{N}$ and a database schema $\mathcal{D}$ is a total function mapping every node of $\mathcal{N}$ to a finite database instance over $\mathcal{D}$. For a node $x \in \mathcal{N}$, we denote $H|^x = \{R(x, \bar{a}) \mid R(\bar{a}) \in H(x)\}$, which is a database instance over $\mathcal{D}^{\mathrm{L}}$. This represents how data is spread over the nodes of a network.

2.2 Datalog with negation

We recall the language Datalog with negation (and nonequalities) as a database query language [4]. We abbreviate this language as Datalog$^\neg$ and we formally define it below. We use a slightly unconventional formalization.

Let **var** be a universe of *variables*, disjoint from **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 φ is a four-tuple

$$(\mathit{head}(\varphi), \mathit{pos}(\varphi), \mathit{neg}(\varphi), \mathit{neq}(\varphi))$$

where $\mathit{head}(\varphi)$ is an atom, $\mathit{pos}(\varphi)$ and $\mathit{neg}(\varphi)$ are sets of atoms, and $\mathit{neq}(\varphi)$ is a set of nonequalities of the form $u \neq v$ where $u, v \in \mathbf{var} \cup \mathbf{dom}$. The components $\mathit{head}(\varphi)$, $\mathit{pos}(\varphi)$ and $\mathit{neg}(\varphi)$ 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*. Importantly, the set $\mathit{neg}(\varphi)$ contains just atoms, not negative literals. Every Datalog$^\neg$ rule φ must have a head, whereas $\mathit{pos}(\varphi)$, $\mathit{neg}(\varphi)$ and $\mathit{neq}(\varphi)$ may be empty. If $\mathit{neg}(\varphi) = \emptyset$ then φ is called *positive*. If all atoms comprising φ are constant-free, then φ is called *constant-free*.

A rule φ may be written in the conventional syntax. For instance, if $\mathit{head}(\varphi) = T(u, v)$, $\mathit{pos}(\varphi) = \{R(u, v)\}$, $\mathit{neg}(\varphi) = \{S(v)\}$ and $\mathit{neq}(\varphi) = \{u \neq v\}$, then we can write φ as

$$T(u, v) \leftarrow R(u, v), \neg S(v), u \neq v.$$

We require rules φ to be *safe* in the sense that the variables occurring in $\mathit{head}(\varphi)$, $\mathit{neg}(\varphi)$ and $\mathit{neq}(\varphi)$ must occur in $\mathit{pos}(\varphi)$. The set of variables of φ is denoted $\mathit{vars}(\varphi)$. If $\mathit{vars}(\varphi) = \emptyset$ then φ is called *ground*, in which case we will consider $\{\mathit{head}(\varphi)\} \cup \mathit{pos}(\varphi) \cup \mathit{neg}(\varphi)$ to be a set of facts.

Let $\mathcal{D}$ be a database schema. A rule φ is said to be *over schema* $\mathcal{D}$ if for each atom $R(u_1, \ldots, u_k) \in \{\mathit{head}(\varphi)\} \cup \mathit{pos}(\varphi) \cup \mathit{neg}(\varphi)$ we have $R^{(k)} \in \mathcal{D}$. Then 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. The database schema that P is over will also be denoted as $\mathit{sch}(P)$. We define $\mathit{idb}(P) \subseteq \mathit{sch}(P)$ to be the database schema consisting of all relations occuring in rule-heads of P. We abbreviate $\mathit{edb}(P) = \mathit{sch}(P) \setminus \mathit{idb}(P)$. An *input* for P is a database instance over $\mathit{sch}(P)$. Note that we allow inputs to already contain facts over $\mathit{idb}(P)$, cf. Section 3.1.

Let P be a Datalog$^\neg$ program. Let I be an instance over $\mathit{sch}(P)$. Let $\varphi \in P$. A *valuation for* φ is a total function $V : \mathit{vars}(\varphi) \to \mathbf{dom}$. We define the *application* of V to an atom $R(u_1, \ldots, u_k)$, 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: variables are replaced by data values and the old data values are left 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 *satisfied* on I if $V(\mathit{pos}(\varphi)) \subseteq I$, $V(\mathit{neg}(\varphi)) \cap I = \emptyset$ and $V(u) \neq V(v)$ for each nonequality $(u \neq v) \in \mathit{neq}(\varphi)$. If V is satisfied on I, then φ is said to *derive* the fact $V(\mathit{head}(\varphi))$.

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 $\mathit{neg}(\varphi)$ are contained in $\mathit{edb}(P)$. Naturally, positive programs are semi-positive.

Let P be a semi-positive Datalog$^\neg$ program. We now give the semantics of P [4]. We define the *immediate consequence operator* T_P that maps each instance J over $\mathit{sch}(P)$ to the instance $J' = J \cup A$ where A contains the facts derived by all possible satisfying valuations for the rules of P on J.

Let I be an instance over $\mathit{sch}(P)$. Consider the infinite sequence I_0, I_1, I_2, etc, with $I_0 = I$ and $I_i = T_P(I_{i-1})$ for $i \geq 1$. We define the output of P on input I, denoted $P(I)$, as $\bigcup_l I_l$; this is the minimal fixpoint of the T_P operator containing I. When I is finite, the fixpoint is finite and can be computed in polynomial time.

2.2.2 Stratified semantics

We now recall the notion of stratified semantics for a Datalog$^\neg$ program [4]. A Datalog$^\neg$ program P is called *syntactically stratifiable* if there is a function $\sigma : \mathit{idb}(P) \to \{1, \ldots, |\mathit{idb}(P)|\}$ such that for each rule $\varphi \in P$, with head predicate T, the following conditions are satisfied:

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

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

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

2.2.3 Stable model semantics

We now recall the notion of stable model semantics [17, 27]. Let P be a Datalog$^\neg$ program and let I be a database instance over $\mathit{sch}(P)$. Let $\varphi \in P$. Let V be a valuation for φ whose image is contained in $\mathit{adom}(I)$. Together, V and φ give rise to a ground rule ψ, that is precisely φ except that each $u \in \mathit{vars}(\varphi)$ is replaced by $V(u)$. We call ψ a *ground rule of* φ *with respect to* I. Let $\mathit{ground}(\varphi, I)$ denote the set of all ground rules of φ we can make with respect to I. The *ground program* of P on input I, denoted $\mathit{ground}(P, I)$, is defined as $\bigcup_{\varphi \in P} \mathit{ground}(\varphi, I)$.

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

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

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

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

2.3 Dedalus programs

We now recall the language Dedalus [7, 8, 20]. Essentially, Dedalus is a fragment of Datalog$^\neg$ extended with dedicated time-relations and choice-operators.

A *choice-operator* is of the form $\mathit{choice}(\langle \bar{x}\rangle, \langle \bar{y}\rangle)$ where $\bar{x}$ and $\bar{y}$ are sequences of variables [22, 27]. Intuitively, this expresses a functional dependency $\bar{x} \to \bar{y}$.

We define the database schema $\mathcal{D}_{\text{time}} = \{\texttt{time}^{(1)}, \texttt{tsucc}^{(2)}\}$. Intuitively, a database instance over $\mathcal{D}_{\text{time}}$ provides a universe of discrete timestamp values with a successor relation. Specifically, the only instance that we will use over this schema will have $\texttt{time} = \mathbb{N}$ and $\texttt{tsucc} = \{(s, s+1) \mid s \in \mathbb{N}\}$.

Now, let $\mathcal{D}$ be a database schema, with relation names disjoint from those in $\mathcal{D}_{\text{time}}$. We first define the general form that all Dedalus rules obey; afterwards we will restrict attention to the three specific cases that are allowed to occur in a program. Formally, a *general Dedalus rule* φ over $\mathcal{D}$ is a *constant-free* Datalog$^\neg$ rule over schema $\mathcal{D}^{\text{LT}} \cup \mathcal{D}_{\text{time}}$ with an additional set $\mathit{cho}(\varphi)$ of choice-operators:

$$(\mathit{head}(\varphi), \mathit{pos}(\varphi), \mathit{neg}(\varphi), \mathit{neq}(\varphi), \mathit{cho}(\varphi))$$

where the components $\mathit{head}(\varphi)$, $\mathit{pos}(\varphi)$, $\mathit{neg}(\varphi)$ and $\mathit{neq}(\varphi)$ are as defined for Datalog$^\neg$ rules and where two restrictions are satisfied: *(i)* the predicate of $\mathit{head}(\varphi)$ must be in $\mathcal{D}^{\text{LT}}$; *(ii)* all variables occurring in $\mathit{cho}(\varphi)$ must be in $\mathit{vars}(\varphi)$, which is simply an extension of the safety requirement.

Next we restrict attention to three specific forms of Dedalus rules. A crucial notation thereto is $\mathbf{B}\{x, s \mid \bar{u}\}$, which denotes any body β that is a sequence of nonequalities and literals over database schema $\mathcal{D}^{\text{LT}}$, such that

- all literals from β have the same variables x and s on the location specifier and timestamp component respectively;
- s occurs *only* in the timestamp component of these literals;[1] and
- the variables that occur in β are precisely x, s and $\bar{u}$.

Consider a general Dedalus rule φ over schema $\mathcal{D}$.

- We call φ *deductive* if it is of the form

$$R(x, s, \bar{v}) \leftarrow \mathbf{B}\{x, s \mid \bar{v}, \bar{w}\}.$$

In this form no relations over $\mathcal{D}_{\text{time}}$ are used. The location specifier and timestamp are the same for all atoms in the body and the head.

- We call φ *inductive* if it is of the form

$$R(x, t, \bar{v}) \leftarrow \mathbf{B}\{x, s \mid \bar{v}, \bar{w}\}, \texttt{tsucc}(s, t).$$

The location specifier in the head is the same as in the body. The timestamp of the head is the successor of the timestamp in the body.

- We call φ *asynchronous* if it is of the form

$$R(y, t, \bar{v}) \leftarrow \mathbf{B}\{x, s \mid \bar{v}, \bar{w}, y\}, \texttt{time}(t), \mathit{choice}(\langle x, s, y, \bar{v}\rangle, \langle t\rangle).$$

The location specifier and timestamp variables of the head may now be different from those in the body. Intuitively, the choice-operator enforces the restriction that the timestamp in the head is functionally determined by the other variables in the head and the location specifier and timestamp in the body.

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 $\mathcal{D}$:

$$\begin{aligned} R(x, s, u, v) &\leftarrow T(x, s, u, v), \neg S(x, s, v). \\ R(x, t, u, v) &\leftarrow T(x, s, u, v), \texttt{tsucc}(s, t). \\ R(y, t, u, v) &\leftarrow T(x, s, u, v), S(x, s, y), \texttt{time}(t), \\ &\qquad \mathit{choice}(\langle x, s, y, u, v\rangle, \langle t\rangle). \end{aligned}$$

We define:

Definition 2.1. A Dedalus program over schema $\mathcal{D}$ is a finite set P of Dedalus rules over $\mathcal{D}$ such that each rule is deductive, inductive, or asynchronous, and such that the set of deductive rules from P is syntactically stratifiable.

We denote $\mathit{sch}(P) = \mathcal{D}$, so not including $\mathcal{D}_{\text{time}}$. Because P is a set of rules, the definitions of $\mathit{idb}(P)$ and $\mathit{edb}(P)$ are like for Datalog$^\neg$ programs.

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

Note that we have defined Dedalus programs to be 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.

2.4 Vector clocks

Vector clocks are a common technique to represent the relative order of events in a distributed system, also known as the *causal order* [12]. Specifically, in our two semantics for Dedalus, vector clocks will allow us to reason about which messages (possibly indirectly) caused other messages to be sent.

Let $\mathcal{N}$ be a network. Formally, a *vector clock* over $\mathcal{N}$ is a function v that maps each node $x \in \mathcal{N}$ to a number in $\mathbb{N}$. It is common to consider a vector clock to be an array, so for $x \in \mathcal{N}$ we will write $v[x]$ to mean $v(x)$.

For each network $\mathcal{N}$, we can define a partial order $\preceq$ on vector clocks over $\mathcal{N}$ as follows: for two vector clocks v_1 and v_2, we write $v_1 \preceq v_2$ iff $v_1[x] \leq v_2[x]$ for all $x \in \mathcal{N}$. For two vector clocks v_1 and v_2 we write $v_1 \prec v_2$ if $v_1 \preceq v_2$ and $v_1 \neq v_2$.

3. OPERATIONAL SEMANTICS

In this section we give an operational semantics for Dedalus. We describe how a network executes a Dedalus program P when a distributed input database is given. The nodes of the network are made active one by one in some arbitrary order, and this continues an infinite number of times. During each active moment, called a *(local) step*, a node receives messages and derives facts by means of simplified deductive, inductive and asynchronous rules. The ordinal of the step of a node is called the *timestamp*, which can be regarded as a local clock value. For technical convenience these ordinals begin at value 0.

3.1 Simplified rules

Let P be a Dedalus program. For a rule $\varphi \in P$ we define the *simplified* version $s(\varphi) = \varphi'$, as follows:

[1] In some earlier presentations of Dedalus [7], s is also allowed to occur in other places within the body. This very powerful feature has been called "temporal entanglement" and allows the simulation of arbitrary Turing machines [9]. We do not consider it here.

- $head(\varphi') = head(\varphi)^{\downarrow}$;
- $pos(\varphi') = \big(pos(\varphi)|_{sch(P)}\big)^{\downarrow}$;
- $neg(\varphi') = \big(neg(\varphi)|_{sch(P)}\big)^{\downarrow}$;
- $neq(\varphi') = neq(\varphi)$; and,
- $cho(\varphi') = \emptyset$.

In words: we drop the use of relations in $\mathcal{D}_{\text{time}}$, the choice-operators and the variables on timestamp components. To illustrate, consider the following (asynchronous) rule φ:

$$T(y,t,u) \;\leftarrow\; R(x,s,y,u), \neg S(x,s,u), x \neq y, \texttt{time}(t), choice(\langle x,s,y,u\rangle, \langle t\rangle).$$

The rule $s(\varphi)$ is then:

$$T(y,u) \leftarrow R(x,y,u), \neg S(x,u), x \neq y.$$

Note that all atoms that make up the simplified rules are over a relation in $sch(P)^{\mathrm{L}}$.

We define $deduc(P)$ to be the Datalog$^{\neg}$ program consisting of precisely all simplified deductive rules of P. Note that this program is syntactically stratifiable because the deductive rules of P are syntactically stratifiable. Similarly, we define $induc(P)$ and $async(P)$ to be respectively the set of all simplified inductive rules of P and the set of all simplified asynchronous rules of P. It is possible that $induc(P)$ and $async(P)$ are not syntactically stratifiable.

Let I be a database instance over $sch(P)^{\mathrm{L}}$. The *output* of $deduc(P)$ on input I, denoted $deduc(P)(I)$, is given by the stratified semantics. Importantly, I is allowed to contain facts over $idb(P)^{\mathrm{L}}$ and the intuition is that these facts are already derived on a previous timestamp (by an inductive rule). The need for this will become clear in Section 3.2.

We define the *output* of $induc(P)$ on input I to be the set of facts derived by the rules of $induc(P)$ for all possible valuations that are satisfied in I, in just one derivation step. Intuitively, there is no fixpoint in this definition because facts derived for the next timestamp are not visible at the current timestamp and can thus not be used again to derive more facts. This output is denoted as $induc(P)(I)$. The output for $async(P)$ on input I is defined similarly to that of $induc(P)$, of course using the rules of $async(P)$ instead of $induc(P)$. The intuition for not requiring a fixpoint for $async(P)$ is that a message fact will arrive at a later moment than when it was sent and will therefore not be used again to derive more facts.

3.2 Transitions and runs

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$.

We define a *configuration* ρ of P on $\mathcal{N}$ to be a pair (s^ρ, b^ρ) where s^ρ is a set of facts over $sch(P)^{\mathrm{L}}$ and b^ρ is a set of pairs $\langle k, \boldsymbol{f}\rangle$ with $k \in \mathbb{N}$ and $\boldsymbol{f}$ is a fact over database schema $idb(P)^{\mathrm{L}}$. The location specifiers of facts in s^ρ or b^ρ must be contained in $\mathcal{N}$, and this way we can uniquely identify the node to which a fact belongs. We call s^ρ the *state* and b^ρ the *(message) buffer* respectively.

Intuitively, a configuration describes the network at a certain point in its evolution: it says something about the state of each node and about the message buffer of each node. The message buffer represents per node the messages that have been sent to that node but that the node has not yet received. Thus, one should not think of the buffer as a queue of already delivered messages, but rather as representing the messages that are still floating around in the network. The number k attached to a fact $\boldsymbol{f}$ in b^ρ, as $\langle k, \boldsymbol{f}\rangle$, indicates that $\boldsymbol{f}$ was sent at global transition number k. This will be made precise below.

The *start configuration* of P on input H, denoted by $start(P,H)$, is the configuration ρ defined by $s^\rho = \bigcup_{x\in\mathcal{N}} H|^x$ and $b^\rho = \emptyset$.

We will now define the transitions to go from one configuration to another, called *global transitions.* They describe how one active node does a local computation step to update its state and to send messages around the network. We call this active node the *recipient* and its local computation step is formalized by means of a *local transition.* Formally, a local transition is a four-tuple $(I, I_{\text{rcv}}, J, J_{\text{snd}})$, also denoted as

$$I, I_{\text{rcv}} \to J, J_{\text{snd}},$$

where

- I is a finite database instance over the schema $sch(P)^{\mathrm{L}}$;
- I_{rcv} is a finite database instance over the schema $idb(P)^{\mathrm{L}}$;
- letting $D = deduc(P)(I \cup I_{\text{rcv}})$, we have
$$J = I|_{edb(P)} \cup induc(P)(D).$$
- $J_{\text{snd}} = async(P)(D)$ with D as above.

Intuitively, instances I and I_{rcv} represent the old state and incoming messages, respectively. The instances J and J_{snd} represent the new state and outgoing messages, respectively. Instance J preserves the input facts of I (having their predicate in $edb(P)$) and it also includes all facts derived inductively from the old state and incoming messages. This represents *mutable state* for the relations in $idb(P)$, where only the facts that are explicitly derived are preserved. Conceptually, the inductive and asynchronous rules are applied after $deduc(P)$ has *completed* the information in $I \cup I_{\text{rcv}}$. This way of defining local transitions is the same as was done for WebDamLog [3].

Note that local transitions are deterministic, in the sense that if $I, I_{\text{rcv}} \to J, J_{\text{snd}}$ and $I, I_{\text{rcv}} \to J', J'_{\text{snd}}$ are local transitions, then $J = J'$ and $J_{\text{snd}} = J'_{\text{snd}}$. Local transitions are computable in polynomial time.

For a set of facts I and a number $i \in \mathbb{N}$ we define the set of pairs $tag(i, I) = \{\langle i, \boldsymbol{f}\rangle \mid \boldsymbol{f} \in I\}$. For a set m of such pairs we define $untag(m) = \{\boldsymbol{f} \mid \exists k : \langle k, \boldsymbol{f}\rangle \in m\}$.

Now we are ready to describe global transitions. Formally, a *global transition* of P on $\mathcal{N}$ with *send-tag* $i \in \mathbb{N}$ is a five-tuple $(\rho_1, x, m, i, \rho_2)$, also denoted as $\rho_1 \xrightarrow[i]{x,m} \rho_2$, where

- ρ_1 and ρ_2 are two configurations of P on $\mathcal{N}$;
- $x \in \mathcal{N}$;
- $m \subseteq b^{\rho_1}$, such that all facts in m have location specifier x;
- there exist database instances J and J_{snd} over $sch(P)^{\mathrm{L}}$ and $idb(P)^{\mathrm{L}}$ respectively such that
$$s^{\rho_1}|^x, untag(m) \to J, J_{\text{snd}}$$

is a local transition of P and

$$s^{\rho_2} = (s^{\rho_1} \setminus s^{\rho_1}|^x) \cup J;$$
$$b^{\rho_2} = (b^{\rho_1} \setminus m) \cup tag(i, \delta)$$

with $\delta = \bigcup_{y \in \mathcal{N}} J_{\text{snd}}|^y$.

In the definition of global transition, we call x the *recipient*, m the set of *delivered messages* and δ the set of *sent messages*. Intuitively, a global transition expresses that the node x receives some messages addressed to it (without the attached number) and locally executes the program P. Possibly new messages are generated this way; the set δ represents the generated messages. Also, note that the location specifiers in J will all be x because *(i)* all location specifiers in $s^{\rho_1}|^x \cup untag(m)$ are x and *(ii)* J is defined only in terms of deductive and inductive rules. For the location specifiers in J_{snd} this does not hold because J_{snd} is also defined in terms of asynchronous rules. The send-tag i is attached to each fact of δ, and the resulting pairs are placed in the message buffer. The facts in J_{snd} with a location specifier not in $\mathcal{N}$ are ignored. The send-tags help us differentiate between multiple instances of the same message that are sent during different global transitions. We could have equivalently modeled the message buffer as a *multiset* of message facts, without any numbers attached to them. The current modelling however turns out to be technically more convenient for talking about when a sent message arrives, as we will see in Section 3.3. Importantly, the send-tags are not visible to the Dedalus program, i.e., to the local transitions.

Note that in a global transition as above, ρ_2 is uniquely determined given ρ_1, x, and m, but because from a given configuration ρ_1 there may be many choices of recipient x and of set m of delivered messages, the global transition system is highly non-deterministic. Now let H be an input distributed database instance for P, over a network $\mathcal{N}$. A *run* $\mathcal{R}$ of P on input H is defined as an infinite sequence of global transitions

$$\rho_0 \xrightarrow[0]{x_0,m_0} \rho_1,\ \rho_1 \xrightarrow[1]{x_1,m_1} \rho_2,\ \rho_2 \xrightarrow[2]{x_2,m_2} \rho_3,\ \ldots$$

where $\rho_0 = start(P, H)$. Note that the next global transition starts from the ending configuration of the previous global transition. We refer to the global transitions with indices $i \in \mathbb{N}$ (where the first transition has index 0). In a run an infinite number of global transitions is always possible because the set of delivered messages may be empty. Note also that $m_0 = \emptyset$ because $b^{\rho_0} = \emptyset$. Finally, note that the local transition happening in each global transition of a run involves finite instances only, because the start configuration given by H is finite (by definition of distributed database instance).

An important aspect of the operational semantics given here is that the predicate of every message fact is simply a relation name of $sch(P)$ itself. This allows the local (simplified) 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.

3.3 Fairness and arrival function

In the literature on process models it is customary to require certain "fairness" conditions on the execution of a system [16, 11, 23].

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on H. Write global transition i of $\mathcal{R}$ as $\rho_i \xrightarrow[i]{x_i,m_i} \rho_{i+1}$. We call $\mathcal{R}$ *fair* if *(i)* every node of $\mathcal{N}$ is the recipient in an infinite number of global transitions of $\mathcal{R}$ and *(ii)* for each $i \in \mathbb{N}$ and for each $\langle k, \boldsymbol{f} \rangle \in b^{\rho_i}$, there exists a $j \geq i$ such that $\langle k, \boldsymbol{f} \rangle \in m_j$.

Intuitively, fairness disallows "starvation": it requires that every node does an infinite number of local computation steps and that every sent message is eventually delivered. In this last condition, it is possible that $j = i$, and in that case $\langle k, \boldsymbol{f} \rangle$ is delivered in the global transition immediately following configuration ρ_i. Also, it follows from the operational semantics that this j is unique for $\langle k, \boldsymbol{f} \rangle$. Hence we will denote j as $\alpha_{\mathcal{R}}(k, \boldsymbol{f})$. This function $\alpha_{\mathcal{R}}$ is called the *arrival function*, and its domain is $sent(\mathcal{R}) = \bigcup_{i \in \mathbb{N}} tag(i, \delta_i)$ where δ_i is the set of sent messages during global transition i. Intuitively, for every global transition index k, for every sent message $\boldsymbol{f} \in \delta_k$, the function $\alpha_{\mathcal{R}}$ maps the pair $(k, \boldsymbol{f})$ to the global transition index j in which $\langle k, \boldsymbol{f} \rangle$ is delivered. An important property of the arrival function is that for $(k, \boldsymbol{f}) \in sent(\mathcal{R})$ we have $\alpha_{\mathcal{R}}(k, \boldsymbol{f}) > k$. 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 global transition. When $\mathcal{R}$ is clear from the context, we write α instead of $\alpha_{\mathcal{R}}$.

Importantly, when we consider a run in this text, this run is assumed to fair, unless explicitly stated otherwise.

4. DECLARATIVE SEMANTICS

In this section we give a declarative semantics to any Dedalus program by using the stable model semantics applied to a pure Datalog$^\neg$ program, obtained from the Dedalus program. The Datalog$^\neg$ program is described in Section 4.1.

4.1 Causality transformation

Let P be a Dedalus program. In the following, we describe how to construct a pure Datalog$^\neg$ program, $pure(P)$, that is based on P. Intuitively, this program enforces the communication, as represented by the asynchronous rules of P, to be *causal* in the sense that a message sent by a node x at local timestamp s cannot directly or indirectly cause a message to arrive in the past of node x, thus *before* local timestamp s. We call $pure(P)$ the *causal version* of P. The key idea in its construction is inspired by the temporal constraints that arise from causally-ordered vector clocks.

We define $pure(P)$ incrementally. All rules we add are constant-free. First, we add to $pure(P)$ all deductive and inductive rules of P. These do not contain choice-operators. Now to augment $pure(P)$ further, we assume without loss of generality that the following relation names do not yet occur in $sch(P) \cup \mathcal{D}_{\text{time}}$: `notZero`, `zero`, $<$, $\leq$, `rcvClock`, `isBehind`, `clock`, `all`, and for each relation name R in $idb(P)$ the relation names R_{snd}, $\texttt{chosen}_R$ and $\texttt{other}_R$. The relations $<$ and $\leq$ are binary and will be written in infix notation. We add the following auxiliary rules to $pure(P)$, just to get the value zero:

$$\texttt{notZero}(t) \leftarrow \texttt{tsucc}(s, t). \qquad (4.1)$$

$$\texttt{zero}(t) \leftarrow \texttt{time}(t), \neg\texttt{notZero}(t). \qquad (4.2)$$

To represent vector clocks, we add the following rules to *pure*(P), for which we give the intuition afterwards:

$$\texttt{rcvClock}(x,s,y,s) \leftarrow \texttt{all}(x), \texttt{all}(y), x \neq y, \texttt{zero}(s). \quad (4.3)$$

$$\texttt{rcvClock}(x,s,x,s') \leftarrow \texttt{all}(x), \texttt{tsucc}(s,s'). \quad (4.4)$$

$$\texttt{rcvClock}(x,s',y,t) \leftarrow \texttt{clock}(x,s,y,t), x \neq y, \texttt{tsucc}(s,s'). \quad (4.5)$$

$$\texttt{isBehind}(x,s,y,t) \leftarrow \texttt{rcvClock}(x,s,y,t), \texttt{rcvClock}(x,s,y,t'), t < t'. \quad (4.6)$$

$$\texttt{clock}(x,s,y,t) \leftarrow \texttt{rcvClock}(x,s,y,t), \neg\texttt{isBehind}(x,s,y,t). \quad (4.7)$$

Here, the relation `all` holds all nodes of the network and will be initialized as an EDB relation. A fact of the form $\texttt{rcvClock}(x,s,y,t)$ expresses that node x at its local time s has a *lower-bound estimate* that node y's local clock has advanced to be at least value t. Possibly $x = y$. There can be multiple `rcvClock`-facts with the first three components and these represent different estimates that node x at its local time s has about y's clock. By contrast, a fact of the form $\texttt{clock}(x,s,y,t)$ expresses that x at its local time s has settled on the largest estimate t about y's clock. So, relation `clock` represents the actual vector clock.

The intuition behind the above rules (which are not stratified) is the following. Rule (4.3) expresses the initialization of clock estimates on node x: the clocks for the other nodes y are assumed to be at least zero. Rule (4.4) expresses how the node x updates its own local component in the vector clock, which is uniformly represented as a `rcvClock`-fact. Intuitively, the use of $\texttt{tsucc}(s,s')$ is to make sure that when x sends a message to itself, the arrival timestamp can be chosen to be strictly larger than the timestamp when it was sent. Rule (4.5) expresses that at the next timestamp, a node x knows its previous vector clock, for nodes different from x. This way the component of a node y in the new vector clock is at least the previous value. Rules (4.6) and (4.7) compute for node x and its local timestamp s the maximum clock estimate t for the node y. Naturally, because in the above rules x is just a variable, the rules compute the vector clock on every node. Importantly, as we will see below, a fact of the form $\texttt{rcvClock}(x,s,y,t)$ can also arise from communication between nodes.

The previous rules are always added to *pure*(P). Finally we have to add rules based on what asynchronous rules occur in P. The construction described next needs to be performed for each asynchronous rule. Using the notations from Section 2.3, suppose we have the following asynchronous rule in P:

$$R(y,t,\bar{v}) \leftarrow \mathbf{B}\{x,s \mid \bar{v},\bar{w},y\}, \texttt{time}(t), \mathit{choice}(\langle x,s,y,\bar{v}\rangle,\langle t\rangle).$$

Based on this rule, we add to *pure*(P) the following rules:

$$R_{\text{snd}}(x,s,y,t,\bar{v}) \leftarrow \mathbf{B}\{x,s \mid \bar{v},\bar{w},y\}, \texttt{all}(y), \texttt{clock}(x,s,y,u), \texttt{time}(t), u \leq t, \texttt{chosen}_R(x,s,y,\bar{v},t). \quad (4.8)$$

$$\texttt{chosen}_R(x,s,y,\bar{v},t) \leftarrow \mathbf{B}\{x,s \mid \bar{v},\bar{w},y\}, \texttt{all}(y), \texttt{clock}(x,s,y,u), \texttt{time}(t), u \leq t, \neg\texttt{other}_R(x,s,y,\bar{v},t). \quad (4.9)$$

$$\texttt{other}_R(x,s,y,\bar{v},t) \leftarrow \mathbf{B}\{x,s \mid \bar{v},\bar{w},y\}, \texttt{all}(y), \texttt{clock}(x,s,y,u), \texttt{time}(t), u \leq t, \texttt{chosen}_R(x,s,y,\bar{v},t'), t \neq t'. \quad (4.10)$$

$$R(y,t,\bar{v}) \leftarrow R_{\text{snd}}(x,s,y,t,\bar{v}). \quad (4.11)$$

$$\texttt{rcvClock}(y,t,z,u) \leftarrow R_{\text{snd}}(x,s,y,t,\bar{v}), \texttt{clock}(x,s,z,u). \quad (4.12)$$

The intuition behind these rules is as follows. Rule (4.8) represents the sending of R-messages that are generated by the original asynchronous rule from P. Relation R_{snd} contains all R-messages that are sent, with some additional information: the sender x, the local clock s of x at the moment of sending, and the local clock value t of recipient node y at the moment of arrival. Intuitively, to compute relation R_{snd}, the sending node x looks at his vector clock to know the best estimate u for the local clock of y. The node x then knows that when it sends a message to node y, then that message will arrive at a local clock value t of y that is *at least* u. If we assume that in relation $\texttt{chosen}_R$ the last component is functionally determined by the set of all other components, then the body of rule (4.8) uses relation $\texttt{chosen}_R$ to select only *one* arrival time for the message. Also, if we assume that relation `all` contains precisely the nodes of a network, then the body atom $\texttt{all}(y)$ allows sending only to valid nodes, and this reflects the operational semantics. Whenever we write "R_{snd}" in this text, it is meant that R is a relation name in *sch*(P) and that R_{snd} is a relation name obtained after applying the above transformation.

Rules (4.9) and (4.10) together enforce the functional dependency $x,s,y,\bar{v} \to t$ expressed by $\mathit{choice}(\langle x,s,y,\bar{v}\rangle,\langle t\rangle)$. These rules simulate choice using non-stratified negation as pioneered by Saccà and Zaniolo [27].

Rule (4.11) represents the arrival of an R-message. The data-tuple $\bar{v}$ becomes part of node y's state for the local arrival time. As the node y works with relation R, it thus transparantly reads the R-messages. This corresponds well to the operational semantics of Section 3.

Rule (4.12) delivers to the recipient y at the local arrival time t the entire vector clock of the sender x at the local sending time s. Intuitively, the idea is that the vector clock of y will incorporate the estimates in node x's vector clock, so that when node y later replies (possibly indirectly) to x, that reply will arrive at a later timestamp of x than s.

If there are multiple asynchronous rules in P with head predicate R, then their corresponding R_{snd}-rules all use the same relation $\texttt{chosen}_R$ in their body. This is to express *set* sending semantics: at any given moment, a node can only send a set of facts, not a multiset. Indeed, the sharing of relation $\texttt{chosen}_R$ makes sure that if the same message fact is generated by multiple asynchronous rules with head relation R_{snd}, then the timestamp of arrival is always the same. If there are multiple asynchronous rules in P with head relation R, then there are also multiple rules in *pure*(P) with head relation $\texttt{chosen}_R$ and $\texttt{other}_R$. The arity of these head relations is always the same because *(i)* the number of variables mentioned in a choice-operator of an asynchronous rule is the number of variables in the head plus two (the variables for the body location specifier and timestamp); and *(ii)* the different asynchronous rules with head relation

R all have the same head arity, namely, the arity of R given by the schema plus two.

Because P is constant-free, $pure(P)$ is as well.

4.2 Input and stable models

Now we define the actual declarative semantics for Dedalus. Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $pure(P)$ be as described in Section 4.1. We now describe an input database instance $decl(H)$ for $pure(P)$.

First we define the set of facts $decl(\mathcal{N}) = \{\texttt{all}(x) \mid x \in \mathcal{N}\}$. Also, let I_{time} be the set of facts consisting of

- $\{\texttt{time}(s), \texttt{tsucc}(s, s+1) \mid s \in \mathbb{N}\}$;
- $\{< (s, s') \mid s, s' \in \mathbb{N},\ s < s'\}$;
- $\{\leq (s, s') \mid s, s' \in \mathbb{N},\ s \leq s'\}$.

We define $decl(H)$ to be the database instance over the schema $edb(P)^{\text{LT}} \cup \{\texttt{all}^{(1)}\} \cup \mathcal{D}_{\text{time}} \cup \{<^{(2)}, \leq^{(2)}\}$ consisting of the following facts:

$$\bigcup_{x \in \mathcal{N}} \bigcup_{s \in \mathbb{N}} (H|^x)^{\uparrow s} \cup decl(\mathcal{N}) \cup I_{\text{time}}.$$

The first term of the above union makes for each node its local input facts available at all local timestamps. The second term gives the set of all nodes. The third term provides relations containing the discrete timestamps and comparison relations. Note that because the set $\mathbb{N}$ is infinite, the instance $decl(H)$ contains infinitely many facts.

We now define the declarative semantics for Dedalus programs:

Definition 4.1. Any stable model of $pure(P)$ on input $decl(H)$ is called a *model* of P on input H.

Recall from Section 4.1 that relation name `all` is not used in the rules of the original Dedalus program P. So, if the rules of P need access to node identifiers, then those must be explicitly provided by some input relations or they must be received from other nodes by means of messages.

4.3 Fairness

Like for the operational semantics, we now give a fairness condition on the declarative semantics. Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a stable model of $pure(P)$ on input $decl(H)$.

We say that M is *fair* if for each pair $(x, s) \in \mathcal{N} \times \mathbb{N}$ the following subset of M is finite:

$$\{R_{\text{snd}}(y, t, z, u, \bar{a}) \in M \mid (z, u) = (x, s) \text{ and } R \in sch(P)\}.$$

Intuitively, this means that only a finite number of messages arrive at node x at *local* timestamp s.

While our main theorem formally justifies this definition, a partial intuition behind it can be found in the operational semantics: in a fair run, every node is the recipient in an infinite number of global transitions and every message is eventually delivered. This prevents the message buffer of every node from becoming infinitely large and thus it only receives finite sets of delivered messages at any of its local timestamps. This intuition is only partial, because the notion of stable model itself also rules out some unfair situations. Indeed, our work leaves open an interesting problem in connection with unfair runs: see the Conclusion.

5. TRACE

In this section we show how to associate with each run of a Dedalus program a set of facts that naturally expresses its computation. This set of facts is called the *trace*. This will allow us to unite the operational and declarative semantics.

As an auxiliary construct, in the first subsection we associate vector clocks with the global transitions of a run.

5.1 Vector clocks

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on input H. We write x_i to denote the recipient during global transition i of $\mathcal{R}$. Below we will define for each global transition index i a vector clock $v_{\mathcal{R}}(i)$. Intuitively, $v_{\mathcal{R}}(i)$ is associated with the local transition of the recipient x_i.

We will need the following auxiliary functions. First, for each global transition index i and $y \in \mathcal{N}$, we define $local_{\mathcal{R}}(i, y)$ to be the number of global transitions in $\mathcal{R}$ that come strictly before i and in which y is the recipient. Thus, if local clock values start at 0, the value $local_{\mathcal{R}}(i, y)$ can be thought of as the local clock value that node y will be at the next time it is the recipient in a global transition $j \geq i$. We abbreviate $local_{\mathcal{R}}(i, x_i)$ as $local_{\mathcal{R}}(i)$. Secondly, for a global transition index i we define $prev_{\mathcal{R}}(i)$ to be the set containing the largest global transition index $j < i$ of $\mathcal{R}$ in which $x_j = x_i$ and if no such j exists, then $prev_{\mathcal{R}}(i)$ is empty. Thus $prev_{\mathcal{R}}(i)$ contains at most one element.

Let α be the arrival function for $\mathcal{R}$. Now we define the vector clocks, by induction on the global transition index i. For the base case (global transition 0), we define $v_{\mathcal{R}}(0)[x_0] = 1$ and $v_{\mathcal{R}}(0)[y] = 0$ for each $y \in \mathcal{N}$ with $y \neq x_0$. For the inductive step, we define $v_{\mathcal{R}}(i)[x_i] = local_{\mathcal{R}}(i) + 1$, and for each $y \in \mathcal{N}$ with $y \neq x_i$ we define

$$v_{\mathcal{R}}(i)[y] = \max\big(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y)\big)$$

where

$$\mu_{\mathcal{R}}(i, y) = \{v_{\mathcal{R}}(k)[y] \mid \exists \boldsymbol{f} : (k, \boldsymbol{f}) \in sent(\mathcal{R}), \alpha(k, \boldsymbol{f}) = i\},$$
$$\pi_{\mathcal{R}}(i, y) = \{v_{\mathcal{R}}(j)[y] \mid j \in prev_{\mathcal{R}}(i)\},$$

and $sent(\mathcal{R})$ is as defined in Section 3.3. In the definition of $\mu_{\mathcal{R}}(i, y)$, we have $k < i$ by definition of α, so the induction is well-founded. Intuitively, if we imagine that during every global transition $j < i$ the node x_j attaches its local vector clock $v_{\mathcal{R}}(j)$ to all messages its sends, then $\mu_{\mathcal{R}}(i, y)$ is the set of all the clock estimates that nodes have about y and that were attached to messages delivered to node x_i during global transition i. Next, $\pi_{\mathcal{R}}(i, y)$ is the singleton set containing the previous vector clock value that x_i had about y if there was a previous transition with recipient x_i, and otherwise $\pi_{\mathcal{R}}(i, y)$ is empty. The addition of the set $\{0\}$ is because $\mu_{\mathcal{R}}(i, y)$ and $\pi_{\mathcal{R}}(i, y)$ can be empty. The reason for calculating the maximum in the definition of $v_{\mathcal{R}}(i)[y]$ is that we want the highest estimate about node y's clock value.

The following property states that the vector clock's estimate about the clock value of other nodes y is never larger than the real clock value of those nodes y.

Proposition 5.1. *Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on input H. Let i be a global transition index. For $y \in \mathcal{N}$, if $x_i \neq y$ then $v_{\mathcal{R}}(i)[y] \leq local_{\mathcal{R}}(i, y)$.*

The proof is in the Appendix (Proposition D.1).

5.2 Trace definition

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on input H. Recall the pure Datalog$^{\neg}$ version *pure*(P) as defined in Section 4.1. In this section we do not really need *pure*(P) itself, but we are going to define for $\mathcal{R}$ a set of facts over the *schema* of *pure*(P) that represents the computation of $\mathcal{R}$. This set is called the *trace* of $\mathcal{R}$.

We write global transition i of $\mathcal{R}$ as $\rho_i \xrightarrow[i]{x_i, m_i} \rho_{i+1}$.

First, let M_{-1} be the set consisting of the following facts:

- $decl(\mathcal{N})$ and I_{time}, as defined in Section 4.2;
- $\{\texttt{rcvClock}(x, 0, y, 0) \mid x, y \in \mathcal{N},\ x \neq y\}$;
- $\{\texttt{notZero}(s) \mid s \in \mathbb{N} \setminus \{0\}\}$;
- $\texttt{zero}(0)$.

Now, by induction, for global transition i of $\mathcal{R}$ we define M_i as the union of M_{i-1} with several different sets of facts, that we individually motivate and describe next. Denote $s = local_{\mathcal{R}}(i)$.

First, we want to describe the *state* for node x_i precisely at local timestamp s. Intuitively, this state consists of the facts over only the schema $sch(P)^{\text{LT}}$ that *(i)* were previously inductively derived and that are now read during global transition i, or that *(ii)* result from freshly arrived messages m_i or that *(iii)* are deductively derived from the previous two. This motivates the addition of the following set of facts to M_i:

$$D_i^{\uparrow s} \tag{5.1}$$

with $D_i = deduc(P)\big(s^{\rho_i}|^{x_i} \cup untag(m_i)\big)$. Note that this corresponds exactly to the operational semantics as defined in Section 3.2.

A second aspect of the trace is the description of the vector clocks. Specifically, for global transition i we have defined in Section 5.1 the vector clock $v_{\mathcal{R}}(i)$, the vector clock that node x_i has during its local timestamp s. Note that we defined these vector clocks solely based on the operational semantics, thus entirely independently of the causality rules (4.3)–(4.7) in *pure*(P). We next add facts that represent the operational vector clocks in terms of the predicates used in the causality rules. Specifically, we add to M_i, in order, the following sets of facts:

$$\{\texttt{rcvClock}(x_i, s, x_i, s+1)\}; \tag{5.2}$$

$$\{\texttt{rcvClock}(x_i, s, y, t) \mid y \in \mathcal{N},\ x_i \neq y,\ t \in \pi_{\mathcal{R}}(i, y)\}; \tag{5.3}$$

$$\begin{aligned}\{\texttt{isBehind}(x_i, s, y, t) \mid\ & \texttt{rcvClock}(x_i, s, y, t) \in M_i,\\ & \texttt{rcvClock}(x_i, s, y, t') \in M_i,\\ & t < t'\};\end{aligned} \tag{5.4}$$

$$\{\texttt{clock}(x_i, s, y, t) \mid y \in \mathcal{N},\ t = v_{\mathcal{R}}(i)[y]\}, \tag{5.5}$$

where $\pi_{\mathcal{R}}(i, y)$ is as defined in Section 5.1.

A third aspect of the trace is representing the message sending in $\mathcal{R}$, again in terms of the predicate names of *pure*(P). Specifically, during global transition i the node x_i produces a set δ_i of sent messages (see the set δ in the definition of global transitions in Section 3.2). Let $\boldsymbol{f} \in \delta_i$, which is a fact of the form $R(y, \bar{a})$. The arrival function α maps the pair $(i, \boldsymbol{f})$ to the global transition index j in which the pair $\langle i, \boldsymbol{f}\rangle$ is delivered. The global transition index j can be uniquely transformed to the local clock $t = local_{\mathcal{R}}(j)$ of the recipient y. Thus the arrival function directly specifies at what local time of y each message of δ_i arrives. Also, it follows from Proposition 5.1 that $v_{\mathcal{R}}(i)[y] \leq t$. The facts $R_{\text{snd}}(x_i, s, y, t, \bar{a})$ and $\texttt{chosen}_R(x_i, s, y, \bar{a}, t)$ over the schema of *pure*(P) capture the sending of $\boldsymbol{f}$, including the arrival time. All other local times $t' \neq t$ of y with $v_{\mathcal{R}}(i)[y] \leq t'$ are not arrival times for $\langle i, \boldsymbol{f}\rangle$ and a fact $\texttt{other}_R(x_i, s, y, \bar{a}, t')$ captures this. Finally, because $\alpha(i, \boldsymbol{f}) = j$, by definition of $v_{\mathcal{R}}(j)$ we have $v_{\mathcal{R}}(i)[z] \leq v_{\mathcal{R}}(j)[z]$ for each $z \in \mathcal{N}$. To model this with facts, we make for each $z \in \mathcal{N}$ a fact $\texttt{rcvClock}(y, t, z, u)$ with $u = v_{\mathcal{R}}(i)[z]$. When we generalize all of the previous fact-constructions to the entire set δ_i, we have to add the following sets to M_i, where $l(i, \boldsymbol{f})$ abbreviates $local_{\mathcal{R}}(\alpha(i, \boldsymbol{f}))$:

$$\{R_{\text{snd}}(x_i, s, y, t, \bar{a}) \mid \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t = l(i, \boldsymbol{f})\}; \tag{5.6}$$

$$\{\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \mid \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t = l(i, \boldsymbol{f})\}; \tag{5.7}$$

$$\begin{aligned}\{\texttt{other}_R(x_i, s, y, \bar{a}, t') \ \mid\ & \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t' \in \mathbb{N},\\ & v_{\mathcal{R}}(i)[y] \leq t',\\ & t' \neq l(i, \boldsymbol{f})\};\end{aligned} \tag{5.8}$$

$$\begin{aligned}\{\texttt{rcvClock}(y, t, z, u) \ \mid\ & \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t = l(i, \boldsymbol{f}),\\ & z \in \mathcal{N},\ u = v_{\mathcal{R}}(i)[z]\}.\end{aligned} \tag{5.9}$$

The attentive reader may wonder whether the `rcvClock`-facts from (5.9) may have an influence on the addition of the `isBehind`-facts from (5.4). But we can show that there is no influence (see Lemma D.4 in the appendix).

Now we have defined M_i for each each global transition index i. The *trace* of $\mathcal{R}$ is defined as:

$$M = \bigcup_{i \in \mathbb{N} \cup \{-1\}} M_i.$$

6. MAIN RESULT

Our main result shows a natural correspondence between the operational and declarative semantics of Dedalus:

Theorem 6.1. *For a Dedalus program P and an input distributed database instance H for P, the set of fair stable models of pure(P) on input decl(H) equals the set of traces of fair runs of P on input H.*

The full proof is in the Appendix, but here we sketch the proofs of the two directions of Theorem 6.1. In Section 6.1 we show that the trace of every fair run is a fair stable model of the declarative semantics and in Section 6.2 we show that for every fair stable model there exists a fair run having that model as its trace.

We use the following notations. Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a set of facts over the schema of *pure*(P). The input instance H will always be clear from the context, and therefore we write $G_M(P)$ to abbreviate the ground program $ground_M(pure(P), decl(H))$.

6.1 Run to stable model

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a fair run of P on input H. Let M be the trace of

$\mathcal{R}$. Using the notations from Section 5.2, we have $M = \bigcup_{i \in \mathbb{N} \cup \{-1\}} M_i$. Let N be the least fixpoint of $G_M(P)$ on input $decl(H)$. We have to show that $M = N$ and that M is fair. We omit the fairness argument in this sketch and only show $M \subseteq N$.

We write global transition i of $\mathcal{R}$ as $\rho_i \xrightarrow[i]{x_i, m_i} \rho_{i+1}$, and we abbreviate $S_i = s^{\rho_i}|^{x_i} \cup untag(m_i)$, $D_i = deduc(P)(S_i)$ and $I_i = induc(P)(D_i)$.

To show $M \subseteq N$ we show $M_i \subseteq N$ by induction on $i = -1, 0, 1, 2$, etc. We omit the sketch for the base case. As induction hypothesis, suppose that $M_{i-1} \subseteq N$ with $i-1 \geq -1$. Thus $i \geq 0$. For the inductive step we have to show that the sets (5.1) through (5.9) defined for global transition index i are included in N. We show this for each set individually. Let $local_{\mathcal{R}}(i)$ and $prev_{\mathcal{R}}(i)$ be as defined in Section 5.1. Denote $s = local_{\mathcal{R}}(i)$.

Consider the set (5.1). It can be shown that $S_i^{\uparrow s} \subseteq N$. Then, since $deduc(P)$ is syntactically stratified, it can be shown by induction on the strata that $D_i^{\uparrow s} \subseteq N$.

The set (5.2) for i is in N because these facts are derived by ground rules of the form (4.4) in $G_M(P)$. To show that the set (5.3) for i is in N, we can first apply the induction hypothesis to $j \in prev_{\mathcal{R}}(i)$ with $local_{\mathcal{R}}(j) = s - 1$ to know that the set (5.5) for j is in N, so for $z \in \mathcal{N}$ we have $\texttt{clock}(x_i, s-1, z, v_{\mathcal{R}}(j)[z]) \in N$. Then ground rules of the form (4.5) in $G_M(P)$ derive in N the facts of set (5.3) for i. The previous two inclusions in N and the induction hypothesis imply that all `rcvClock`-facts in M_i are in N. Now, positive ground rules of the form (4.6) derive in N the set (5.4) for i.

Now we show inclusion of the set (5.5) for i. Let $y \in \mathcal{N}$ and let $t = v_{\mathcal{R}}(i)[y]$. We must show that $\texttt{clock}(x_i, s, y, t) \in N$. It can be shown that $\texttt{rcvClock}(x_i, s, y, t) \in N$ and that there is no value $u \in \mathbb{N}$ such that $\texttt{rcvClock}(x_i, s, y, u) \in M_i$ with $t < u$. Then by definition of the set (5.4) for i, we have $\texttt{isBehind}(x_i, s, y, t) \notin M_i$. It can be shown that the set (5.4) for i is the only part of M where we add `isBehind`-facts with first components x_i and s. Thus $\texttt{isBehind}(x_i, s, y, t) \notin M$ and, based on the form (4.7), the following ground rule is in $G_M(P)$ and it derives $\texttt{clock}(x_i, s, y, t) \in N$:

$$\texttt{clock}(x_i, s, y, t) \leftarrow \texttt{rcvClock}(x_i, s, y, t).$$

Now we show inclusion in N of the sets (5.6) to (5.9) of global transition i. For $\boldsymbol{f} \in \delta_i$ we make the following reasoning, with δ_i the set of sent messages during global transition i. Fact $\boldsymbol{f}$ is of the form $R(y, \bar{a})$. Denote $k = \alpha(i, \boldsymbol{f})$ and $t = local_{\mathcal{R}}(k)$. It can be shown that $v_{\mathcal{R}}(i)[y] \leq t$. Since $\boldsymbol{f} \in \delta_i$ there is a simplified rule φ of $async(P)$ and valuation V that have derived $\boldsymbol{f}$. Also, by definition of the set (5.8) for i, we have $\texttt{other}_R(x_i, s, y, \bar{a}, t) \notin M$. Then for inclusion of the set (5.7) of i in N, there is a ground rule of the form (4.9), based on φ and V, that derives $\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$. For inclusion of the set (5.8) of i in N, because $\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$, there are ground rules of the form (4.10) in $G_M(P)$, based on φ and V, that derive $\texttt{other}_R(x_i, s, y, \bar{a}, t')$ for $t' \in \mathbb{N}$ with $v_{\mathcal{R}}(i)[y] \leq t'$ and $t' \neq t$. The inclusion of the sets (5.6) and (5.9) of i can be shown because $\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$, and we omit the sketch.

The inclusion $N \subseteq M$ is shown by induction of the (infinite) fixpoint computation of N, and arguing separately for each predicate of $pure(P)$.

6.2 Stable model to run

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a fair stable model of $pure(P)$ on input $decl(H)$. By definition $M = G_M(P)(decl(H))$.

The direction already shown in Section 6.1 is perhaps the most intuitive direction because we only have to show that the concrete trace is actually a stable model. The core part of the direction shown here is to construct a run out of M.

6.2.1 Local vector clocks

A first step towards constructing a run is understanding how we can (causally) order the computation events that are represented by the facts in M. To this purpose, we now show that for $x, y \in \mathcal{N}$ and $s \in \mathbb{N}$ there is precisely one value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$. This is a crucial insight in the clock information represented by M, and it depends on the fairness assumption on M.

Firstly, it can be shown that for each $\texttt{clock}(a, b, c, d) \in M$ and $\texttt{rcvClock}(a, b, c, d) \in M$ we have $a, c \in \mathcal{N}$ and $b, d \in \mathbb{N}$. We now show that there is *at most* one value t such that $\texttt{clock}(x, s, y, t) \in M$. Suppose there are $\texttt{clock}(x, s, y, t) \in M$ and $\texttt{clock}(x, s, y, t') \in M$ with $t \neq t'$. These facts are derived by ground rules of the form (4.7), and thus $\texttt{rcvClock}(x, s, y, t) \in M$ and $\texttt{rcvClock}(x, s, y, t') \in M$. Without loss of generality, we may assume that $t < t'$. Then a ground rule of the form (4.6) derives $\texttt{isBehind}(x, s, y, t) \in M$. But then the following ground rule, based on the rule (4.7), cannot exist in $G_M(P)$:

$$\texttt{clock}(x, s, y, t) \leftarrow \texttt{rcvClock}(x, s, y, t).$$

Besides (4.7) there is no other rule in $pure(P)$ to derive `clock`-facts, so $\texttt{clock}(x, s, y, t) \notin M$, a contradiction.

Now we show that there is *at least* one value t such that $\texttt{clock}(x, s, y, t) \in M$. It can be shown that there is at least one value u such that $\texttt{rcvClock}(x, s, y, u) \in M$. Suppose that $\texttt{clock}(x, s, y, u) \notin M$. Then, based on rule (4.7), the following ground rule can not be available in $G_M(P)$, because otherwise $\texttt{clock}(x, s, y, u) \in M$:

$$\texttt{clock}(x, s, y, u) \leftarrow \texttt{rcvClock}(x, s, y, u).$$

This implies that $\texttt{isBehind}(x, s, y, u) \in M$. The only rule in $pure(P)$ to derive `isBehind`-facts is (4.6), so the existence of $\texttt{isBehind}(x, s, y, u) \in M$ implies that there is some value u' such that $\texttt{rcvClock}(x, s, y, u') \in M$ and $u < u'$. Now similarly, $\texttt{clock}(x, s, y, u') \notin M$ would again imply that there is yet another fact $\texttt{rcvClock}(x, s, y, u'') \in M$ such that $u' < u''$. Importantly, if there would only be a finite number of `rcvClock`-facts with first two components x and s, then we are bound to eventually find the existence of a value t such that $\texttt{rcvClock}(x, s, y, t) \in M$ and $\texttt{isBehind}(x, s, y, t) \notin M$. Then the following ground rule, based on the form (4.7), exists in $G_M(P)$ and it derives $\texttt{clock}(x, s, y, t) \in M$:

$$\texttt{clock}(x, s, y, t) \leftarrow \texttt{rcvClock}(x, s, y, t).$$

We show that there actually are only a finite number of `rcvClock`-facts with first two components x and s.

Ground rules of the forms (4.3), (4.4), (4.5) can together produce only a finite number of `rcvClock`-facts with first two components x and s because *(i)* in $decl(H)$ there are only a finite number of `all`-facts and the `tsucc`-facts form a chain, and *(ii)* because for $y \in \mathcal{N}$ there is at most one value t such that $\texttt{clock}(x, s, y, t) \in M$ (see above). Also, because M is fair, there are only a finite number of ground rules of the form (4.12) that derive `rcvClock`-facts with first two components x and s.

Let $L = \mathcal{N} \times \mathbb{N}$. The intuitive meaning of a pair $(x, s) \in L$ is that s is a local timestamp of the node x. So, L contains per node all local timestamps. Now, for $(x, s) \in L$ we define the *(local) vector clock* over $\mathcal{N}$ associated with node x at local timestamp s, denoted $v_M(x, s)$, as follows: for $y \in \mathcal{N}$, we define $v_M(x, s)[y] = t$ such that $\texttt{clock}(x, s, y, t) \in M$, which we have just shown to be unique. It can be shown that for $(x, s) \in L$ and $(y, t) \in L$ that if $(x, s) \neq (y, t)$ then $v_M(x, s) \neq v_M(y, t)$.

6.2.2 Construction of run

In this section we show how to construct from M a fair run $\mathcal{R}$ of P on input H, whose trace is again M.

Let L be as previously defined. We can define a partial order $\preceq_L$ on L as follows: $(x, s) \preceq_L (y, t)$ iff $v_M(x, s) \preceq v_M(y, t)$. This relation has the intuition of a happens-before relation [12], but the novelty is that it comes from a purely declarative model M. Choose a total order $\leq_L$ on L that respects $\preceq_L$, in the sense that $(x, s) \leq_L (y, t)$ if $(x, s) \preceq_L (y, t)$. Ordering the elements of L according to the order $\leq_L$ gives us a sequence C.

Let $\mathit{ord} : L \to \mathbb{N}$ denote the function that maps a pair $(x, s) \in L$ to its ordinal in the sequence C. Ordinals start at 0. We can uniquely define the sequence of nodes x_0, x_1, x_2, etc such that for each $i \in \mathbb{N}$ there exists a value $s_i \in \mathbb{N}$ such that $\mathit{ord}(x_i, s_i) = i$. Intuitively, these nodes, in order, will be the recipients during the global transitions of our constructed run.

Let $\mathit{local}_C : \mathbb{N} \times \mathcal{N} \to \mathbb{N}$ be the function that maps a pair (i, x) to the size of the set $\{(x, s) \in L \mid \mathit{ord}(x, s) < i\}$. So, $\mathit{local}_C(i, x)$ is the number of pairs of L in which x occurs that are ordered strictly before position i in C. Intuitively, if i is regarded to be a global transition index, the number $\mathit{local}_C(i, x)$ is the timestamp of the *local* transition that the node x will perform next, during a global transition with index $j \geq i$. Intuitively, this correponds to the function $\mathit{local}_{\mathcal{R}}$ in section 5.1.

Next, based on sequence C, we will define a sequence of tuples, where each tuple resembles a global transition. First we define the sequence of configurations entailed by C. We define the function *state* that maps $(x, s) \in L$ to the set of facts $H|^x \cup (M^{\text{ind}}|^{x,s})^{\downarrow}$, where M^{ind} denotes the restriction of M to the facts $\boldsymbol{f}$ over schema $\mathit{sch}(P)^{\text{LT}}$ for which there exists a (positive) *inductive* ground rule $\psi \in G_M(P)$ with head $\boldsymbol{f}$ and whose body is true on M. Also, we define the function *pairs* that maps a set I of R_{snd}-facts with R a relation in $\mathit{sch}(P)$ to the set

$$\mathit{pairs}(I) = \{\langle \mathit{ord}(x, s), R(y, \bar{a})\rangle \mid \exists t : R_{\text{snd}}(x, s, y, t, \bar{a}) \in I\}.$$

Formally, for $i \in \mathbb{N}$ we define configuration ρ_i of P on input H as follows:

- $s^{\rho_i} = \bigcup_{x \in \mathcal{N}} \mathit{state}(x, \mathit{local}_C(i, x))$;
- $b^{\rho_i} = \mathit{pairs}(\text{buf}_i)$ with

$$\text{buf}_i = \{R_{\text{snd}}(y, t, z, u, \bar{a}) \in M \mid \mathit{ord}(y, t) < i \leq \mathit{ord}(z, u)\}.$$

Using this definition, we obtain a sequence of configurations ρ_0, ρ_1, ρ_2, etc. Now we define the sequence of tuples, one tuple per index $i \in \mathbb{N}$. For $i \in \mathbb{N}$, the tuple τ_i is defined as $(\rho_i, x_i, m_i, i, \rho_{i+1})$, where $m_i = \mathit{pairs}(\text{deliv}_i)$ with

$$\text{deliv}_i = \{R_{\text{snd}}(y, t, z, u, \bar{a}) \in M \mid \mathit{ord}(z, u) = i\}.$$

Let us denote this sequence of tuples as $\mathcal{R}$. We next show that $\mathcal{R}$ is a fair run of P on input H, having M as its trace.

6.2.3 Correctness

Let $\mathcal{R}$ be as previously defined. First, it can be shown (argument omitted) that $\rho_0 = \mathit{start}(P, H)$. Now let $i \in \mathbb{N}$. We must show that τ_i is a valid global transition of P on input H. It can be shown that $m_i \subseteq b^{\rho_i}$. Therefore, there exists a configuration ρ such that $\rho_i \xrightarrow[i]{x_i, m_i} \rho$, and this ρ is unique. The (lengthy) proof that $\rho_{i+1} = \rho$ is omitted from this sketch.

Finally, we can show that $\mathcal{R}$ is fair as follows. Let $i \in \mathbb{N}$. Firstly, every node $x \in \mathcal{N}$ is the recipient in an infinite number of global transitions because there are an infinite number of pairs in L with first component x. Now we show that every message is eventually delivered. Let $\langle k, \boldsymbol{f}\rangle \in b^{\rho_i}$. Fact $\boldsymbol{f}$ is of the form $R(y, \bar{a})$. By definition of b^{ρ_i}, $\langle k, \boldsymbol{f}\rangle \in b^{\rho_i}$ means that there is some fact $R_{\text{snd}}(x, s, y, t, \bar{a}) \in M$ such that $k = \mathit{ord}(x, s) < i \leq \mathit{ord}(y, t)$. Denote $j = \mathit{ord}(y, t)$, so $i \leq j$. By definition, $m_j = \mathit{pairs}(\text{deliv}_j)$ with

$$\text{deliv}_j = \{R_{\text{snd}}(x', s', y', t', \bar{b}) \in M \mid \mathit{ord}(y', t') = j\}.$$

Therefore, $R_{\text{snd}}(x, s, y, t, \bar{a}) \in \text{deliv}_j$ and $\langle k, \boldsymbol{f}\rangle \in m_j$. We conclude that the run $\mathcal{R}$ is fair.

Similarly to the sketch in Section 6.1, we can show that the trace of $\mathcal{R}$ is included in M. Then, because the trace is a stable model itself, it can be shown that the trace actually equals M.

7. CONCLUSION AND FUTURE WORK

We have proven that stable models can be used to formally reason about distributed programs, in addition to more operational formalisms.

One technical issue concerns our restriction to fair runs and models. Intriguingly, our current proof approach really relies on fairness; the aspects related to fairness cannot be simply removed from the proof to obtain the corresponding theorem not restricted to fair runs and models. While fairness is certainly a desirable property, it would still be good to have a characterization of the set of traces of all possible runs, including unfair ones. This is an interesting topic for further work.

Another interesting question concerns the necessity of the causality rules (4.3)–(4.7) present in the pure Datalog$^{\neg}$ version $\mathit{pure}(P)$ of any Dedalus program P used to define the declarative semantics. To investigate the necessity of the causality rules, one may omit them from $\mathit{pure}(P)$; call this the "noncausal" variant of $\mathit{pure}(P)$. Certainly the causality rules are necessary in the sense that there exist programs for which the noncausal pure version has stable models that do not correspond to any run. But in practice, one is mainly interested in programs that have a deterministic result; this can easily be formalized in the operational semantics by requiring, for some designated output relations, that their content becomes eventually the same in every possible fair run (eventual consistency [28, 20, 9, 10]). By our theorem, if P is eventually consistent, all stable models of $\mathit{pure}(P)$ will agree on these output relations. We conjecture that this does not hold for all stable models of the noncausal variant. A natural next step is to consider syntactic restrictions on programs, such as negation-free programs, where consistency is guaranteed [3]. One side of the CRON conjecture [20] is that for such programs, the causal rules are redundant. Settling this conjecture is another interesting topic for further work.

Dedalus is a language with the promise that many distributed computations can be programmed in it (e.g., [5]). Until now, claims on expressive power and other properties of the language could not be rigorously proved correct, due to the lack of a formal semantics. We hope that our work encourages further exploration of the potential for logic to illuminate issues in distributed computing, via both declarative semantics and operational formalisms.

8. REFERENCES

[1] K. Aberer, editor. *Special Section on Peer to Peer Data Management*, volume 32(3) of *SIGMOD Record*. ACM, September 2003.

[2] S. Abiteboul, O. Benjelloun, and T. Milo. The Active XML project: an overview. *The VLDB Journal*, 17(5):1019–1040, 2008.

[3] 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 283–292. ACM Press, 2011.

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

[5] P. Alvaro, T. Condie, N. Conway, J.M. Hellerstein, and R. Sears. I do declare: consensus in a logic language. *SIGOPS Review*, 43(4):25–30, 2010.

[6] P. Alvaro, N. Conway, J.M 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.

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

[8] P. Alvaro, W.R. Marczak, et al. Dedalus: Datalog in time and space. In de Moor et al. [14]. To appear.

[9] 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.

[10] T.J. Ameloot and J. Van den Bussche. Deciding eventual consistency for a simple class of relational transducer networks. In *Proceedings 15th International Conference on Database Theory*, 2012. To appear.

[11] K.R. Apt, N. Francez, and S. Katz. Appraising fairness in languages for distributed programming. *Distributed Computing*, 2(4):226–241, 1988.

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

[13] D. Calvanese, G. De Giacomo, M. Lenzerini, and R. Rosati. Logical foundations of peer-to-peer data integration. In *Proceedings 23rd ACM Symposium on Principles of Database Systems*, pages 241–251. ACM Press, 2004.

[14] O. de Moor, G. Gottlob, T. Furche, and A. Sellers, editors. *Datalog Reloaded: Proceedings Datalog 2.0 Workshop, 2010*, volume 6702 of *Lecture Notes in Computer Science*, 2011. To appear.

[15] 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.

[16] N. Francez. *Fairness*. Springer-Verlag, 1986.

[17] M. Gelfond and V. Lifschitz. The stable model semantics for logic programming. In R.A. Kowalski and K.A. Bowen, editors, *Proceedings 5th International Conference and Symposium on Logic Programming*, pages 1070–1080. MIT Press, 1988.

[18] 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.

[19] J.M. Hellerstein. Datalog redux: experience and conjecture. PODS 2010 keynote.

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

[21] S.S. Huang, T.J. Green, and B.T. Loo. Datalog and emerging applications: an interactive tutorial. In *Proceedings 2011 ACM SIGMOD International Conference on Management of Data*, pages 1213–1216. ACM Press, 2011.

[22] R. Krishnamurthy and S.A. Naqvi. Non-deterministic choice in Datalog. In C. Beeri, J.W. Schmidt, and U. Dayal, editors, *Proceedings 3rd International Conference on Data and Knowledge Bases*, pages 416–424. Morgan Kaufmann, 1988.

[23] L. Lamport. Fairness and hyperfairness. *Distributed Computing*, 13(4):239–245, 2000.

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

[25] 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.

[26] V. Nigam et al. An operational semantics for Network Datalog. In *Third International Workshop on Logics, Agents, and Mobility*, 2010.

[27] 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.

[28] W. Vogels. Eventual consistency. *Communications of the ACM*, 52(1):40–44, 2009.

[29] D. Zinn, T.J. Green, and B. Ludaescher. Win-move is coordination-free. In *Proceedings 15th International Conference on Database Theory*, 2012. To appear.

APPENDIX

The appendix contains the proofs of all the results and further needed notations. Everything is grouped into coherent sections that correspond to the organization of the paper.

A. GENERAL NOTATIONS

For technical convenience, we assume for every Dedalus program that some fixed syntactic stratification is used for the deductive rules. A stratum is indicated by its stratum number (≥ 1). For technical convenience we assume that a stratum 0 exists that contains no rules. Let P be a Dedalus program. Recall the notation $\mathit{deduc}(P)$ from Section 3.1. We write $\mathit{deduc}_k(P)$ to denote the rules of $\mathit{deduc}(P)$ whose stratum number is smaller than or equal to k.

We sometimes write $\mathbf{B}$ to denote a sequence of nonequalities and literals over $\mathit{sch}(P)^{\mathrm{LT}}$ that all have the same location specifier and timestamp. The exact values used in $\mathbf{B}$ are not important in this notation.

Let H be an input distributed database instance for P. Let M be a set of facts over the schema of $\mathit{pure}(P)$. To obtain a simpler notation, we write $G_M(P, H)$ to abbreviate $\mathit{ground}_M(\mathit{pure}(P), \mathit{decl}(H))$. If H is clear from the context, we write $G_M(P)$.

B. OPERATIONAL SEMANTICS

To avoid repetitive definitions at the beginning of the lemmas, consider the following setting to which we can refer:

Setting B.1. Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on input H. □

B.1 Notations about runs

Consider Setting B.1. Let i be a global transition of $\mathcal{R}$. When $\mathcal{R}$ is known from the context, we use the following notations:

- The global transition i of $\mathcal{R}$ is denoted as $\rho_i \xrightarrow[i]{x_i, m_i} \rho_{i+1}$.
- We denote $S_i = s^{\rho_i}|^{x_i} \cup \mathit{untag}(m_i)$, $D_i = \mathit{deduc}(P)(S_i)$, $I_i = \mathit{induc}(P)(D_i)$ and $\delta_i = \bigcup_{y \in \mathcal{N}} \big(\mathit{async}(P)(D_i)\big)|^y$. Here, S, D and I stand for "state", "deductive" and "inductive" respectively. Also, δ_i are the messages effectively sent during global transition i, because they have a valid recipient (a node in $\mathcal{N}$). By definition of the semantics of $\mathit{deduc}(P)(S_i)$, we have $S_i \subseteq \mathit{deduc}(P)(S_i)$.

B.2 General properties

Lemma B.2. *Consider Setting B.1. Let ρ_i be a configuration of $\mathcal{R}$. For $x \in \mathcal{N}$ we have $H|^x \subseteq s^{\rho_i}|^x$.*

Proof. We show this by induction on i. For the base case ($i = 0$) we have $\rho_0 = \mathit{start}(P, H)$ and then by definition of $\mathit{start}(P, H)$ we have $H|^x \subseteq s^{\rho_0}|^x$ for $x \in \mathcal{N}$. For the induction hypothesis, we assume that $H|^x \subseteq s^{\rho_{i-1}}|^x$ for $x \in \mathcal{N}$, with $i - 1 \geq 0$. For the inductive step, we show that $H|^x \subseteq s^{\rho_i}|^x$ for $x \in \mathcal{N}$. Global transition $i-1$ is denoted as $\rho_{i-1} \xrightarrow[i-1]{x_{i-1}, m_{i-1}} \rho_i$. Firstly, consider $y \in \mathcal{N} \setminus \{x_{i-1}\}$. By definition of global transition we have $s^{\rho_{i-1}}|^y = s^{\rho_i}|^y$ and by now applying the induction hypothesis, we have $H|^y \subseteq s^{\rho_{i-1}}|^y = s^{\rho_i}|^y$. Secondly, by definition of global transition there are instances J and J_{snd} over $\mathit{sch}(P)^{\mathrm{L}}$ such that

$$s^{\rho_{i-1}}|^{x_{i-1}}, \mathit{untag}(m_{i-1}) \rightarrow J, J_{\mathrm{snd}}$$

is a valid local transition of P. Now, we have $H|^{x_{i-1}} \subseteq s^{\rho_{i-1}}|^{x_{i-1}}$ by the induction hypothesis and by definition of local and global transition we then have $H|^{x_{i-1}} \subseteq J \subseteq s^{\rho_i}|^{x_{i-1}}$, because local transitions preserve facts over $\mathit{edb}(P)^{\mathrm{LT}}$. □

Lemma B.3. *Consider Setting B.1. Consider a pair $(x, s) \in \mathcal{N} \times \mathbb{N}$. There is precisely one global transition i in $\mathcal{R}$ for which $x_i = x$ and $\mathit{local}_{\mathcal{R}}(i) = s$.*

Proof. Because $x \in \mathcal{N}$, $s \geq 0$ and because of fairness, there is at least one global transition i in $\mathcal{R}$ for which $x_i = x$ and $s = \mathit{local}_{\mathcal{R}}(i)$. But for another global transition $j \neq i$ with $x_j = x$, we must have $\mathit{local}_{\mathcal{R}}(j) \neq \mathit{local}_{\mathcal{R}}(i)$ by definition of the function $\mathit{local}_{\mathcal{R}}$. Thus i is unique. □

Lemma B.4. *Consider Setting B.1. Let $i \in \mathbb{N}$ be a global transition index. We have that δ_i is finite.*

Proof. We show by induction on $i \in \mathbb{N}$ that s^{ρ_i} and b^{ρ_i} are finite. This implies that δ_i is finite by the semantics of local transitions: on finite inputs, the programs $\mathit{deduc}(P)$ and $\mathit{async}(P)$ produce finite outputs.

For the base case ($i = 0$) the property holds because $\rho_0 = \mathit{start}(P, H)$. For the induction hypothesis, we assume that $s^{\rho_{i-1}}$ and $b^{\rho_{i-1}}$ are finite, with $i - 1 \geq 0$. Now consider global configuration $i - 1$: $\rho_{i-1} \xrightarrow[i-1]{x_{i-1}, m_{i-1}} \rho_i$. For the inductive step, we show that s^{ρ_i} and b^{ρ_i} are finite. Consider the set $S_{i-1} = s^{\rho_{i-1}}|^{x_{i-1}} \cup \mathit{untag}(m_{i-1})$. The set S_{i-1} is finite because by the induction hypothesis $s^{\rho_{i-1}}$ is finite and $b^{\rho_{i-1}}$ is finite, and by definition $m_{i-1} \subseteq b^{\rho_{i-1}}$. This implies that $D_{i-1} = \mathit{deduc}(P)(S_i)$, $\mathit{induc}(P)(D_{i-1})$ and $\mathit{async}(P)(D_{i-1})$ are finite. Thus δ_{i-1} is finite. By the semantics of global transitions, it follows that now s^{ρ_i} and b^{ρ_i} are finite. □

C. DECLARATIVE SEMANTICS

C.1 About Dedalus ground programs

Let P be a Dedalus program. Let H be an input distributed database instance for P. For a set of facts M over the schema of $\mathit{pure}(P)$, the ground rules of $G_M(P)$ that are based on the original rules of P itself can be nicely divided into deductive, inductive and asynchronous rules. The *deductive ground rules* are recognizable as the rules with their head predicate in $\mathit{sch}(P)^{\mathrm{LT}}$ and in which the location specifier and timestamp of the head are the same as in the body, and in which no relations of $\mathcal{D}_{\mathrm{time}}$ are used. For a deductive ground rule $\psi \in G_M(P)$ we define its stratum number $\mathit{stratum}(\psi)$ to be the stratum number of the head relation. The *inductive ground rules* are recognizable as the rules with their head predicate in $\mathit{sch}(P)^{\mathrm{LT}}$, in which the timestamp of the head is the successor of the timestamp in the body, in which the location specifier is the same in the head and the body, and in which one `tsucc`-atom occurs in the body. Finally, the *asynchronous ground rules* are recognizable as the rules with a head predicate of the form R_{snd} with R a relation name of $\mathit{sch}(P)$ and in which some $\mathtt{chosen}_R$-atom occurs in the body.

C.2 Further notations

Let P be a Dedalus program. Let H be an input distributed database instance. Let M be a set of facts over the schema of $\mathit{pure}(P)$. Let $G_M(P)$ denote the ground program based on M and input I.

Let $M|^{x,s}$ denote the restriction of M to the facts $\boldsymbol{f}$ over schema $\mathit{sch}(P)^{\mathrm{LT}}$ that have location specifier x and timestamp s.

Let M^{ind} denote the restriction of M to the facts $\boldsymbol{f}$ over schema $\mathit{sch}(P)^{\mathrm{LT}}$ for which there exists an *inductive* ground rule $\psi \in G_M(P)$ with head $\boldsymbol{f}$ and whose body is true on M (formally, $\mathit{pos}(\psi) \subseteq M$ and the nonequalities hold).

Let $M^{\blacktriangle}$ be defined as follows:

$$\begin{aligned} M^{\blacktriangle} &= M|_{\mathit{edb}(P)^{\mathrm{LT}}} \cup \\ &\quad M^{\mathrm{ind}} \cup \\ &\quad \{R(x,s,\bar{a}) \mid \exists y,t : R_{\mathrm{snd}}(y,t,x,s,\bar{a}) \in M\}. \end{aligned}$$

Intuitively, $M^{\blacktriangle} \subseteq M$ contains the input facts, the inductively derived facts and the received facts, all over the schema $\mathit{sch}(P)^{\mathrm{LT}}$.

Let k be a stratum number. We write M_k to denote the *union* of $M^{\blacktriangle}$ with the restriction of M to all facts $\boldsymbol{f}$ over schema $\mathit{sch}(P)^{\mathrm{LT}}$ for which there exists a *deductive* ground rule $\psi \in G_M(P)$ of stratum k, with head $\boldsymbol{f}$ and whose body is true on M (formally, $\mathit{pos}(\psi) \subseteq M$ and the nonequalities are satisfied). This second group of facts can have an overlap with $M^{\blacktriangle}$. To rephraze, M_k contains the input, the inductively derived facts, the received facts and finally, the facts that are derived by deductive rules whose stratum is less than or equal to k. The intuition about M_k is that $M^{\blacktriangle}$ contains the inputs for the deductive reasoning, together with facts derived by strata up to and including stratum k.

C.3 General properties

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a set of facts over the schema of $\mathit{pure}(P)$. A fact $\boldsymbol{f} \in M$ is called *well-formed* when the following conditions are satisfied:

- if $\boldsymbol{f}$ is over $\mathit{sch}(P)^{\mathrm{LT}}$, then the location specifier of $\boldsymbol{f}$ is in $\mathcal{N}$ and the timestamp of $\boldsymbol{f}$ is in $\mathbb{N}$;
- if $\boldsymbol{f}$ is of the form $\mathtt{rcvClock}(x,s,y,t)$, $\mathtt{isBehind}(x,s,y,t)$ or $\mathtt{clock}(x,s,y,t)$, then $x \in \mathcal{N}$, $y \in \mathcal{N}$, $s \in \mathbb{N}$ and $t \in \mathbb{N}$;
- if $\boldsymbol{f}$ is of the form $R_{\mathrm{snd}}(x,s,y,t,\bar{a})$, $\mathtt{chosen}_R(x,s,y,\bar{a},t)$ or $\mathtt{other}_R(x,s,y,\bar{a},t)$ with R a relation name in $\mathit{sch}(P)$, then $x \in \mathcal{N}$, $y \in \mathcal{N}$, $s \in \mathbb{N}$ and $t \in \mathbb{N}$;
- if $\boldsymbol{f}$ is of the form $\mathtt{zero}(t)$ or $\mathtt{notZero}(t)$, then $t \in \mathbb{N}$;
- if $\boldsymbol{f}$ is of the form $\mathtt{all}(x)$ then $x \in \mathcal{N}$;
- if $\boldsymbol{f}$ is of the form $\mathtt{time}(s)$ then $s \in \mathbb{N}$;
- if $\boldsymbol{f}$ is of the form $\mathtt{tsucc}(s,t)$, $<(s,t)$ or $\leq(s,t)$ then $s,t \in \mathbb{N}$.

We call M *well-formed* if all facts in M are well-formed. Intuitively, this property ensures that we use nodes and timestamps on places were we expect nodes and timestamps respectively.

Lemma C.1. *Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a set of facts over the schema of $\mathit{pure}(P)$. Let N be the output of $G_M(P)$ on input $\mathit{decl}(H)$. We have that N is well-formed.*

Proof. We may assume that N is calculated by deriving one fact at a time. So we have a sequence $N_0 \subseteq N_1 \subseteq N_2 \subseteq \ldots$ with $N_0 = \mathit{decl}(H)$ and $N_\infty = N$. We show by induction on i that N_i is well-formed. For the base case ($i = 0$), the property immediately holds because $\mathit{decl}(H)$ is well-formed by construction. For the induction hypothesis, assume that N_{i-1} with $i - 1 \geq 0$ is well-formed. We show that N_i is well-formed. Let $\boldsymbol{f}$ be the fact that is added to N_{i-1} in order to obtain N_i. Let ψ be a ground rule of $G_M(P)$ that has derived $\boldsymbol{f}$. We have $\mathit{pos}(\psi) \subseteq N_{i-1}$ and thus, by applying the induction hypothesis, we know that $\mathit{pos}(\psi)$ is well-formed. We now consider only the cases where $\boldsymbol{f}$ is over $\mathit{idb}(\mathit{pure}(P))$:

- Suppose $\boldsymbol{f}$ is over $\mathit{sch}(P)^{\mathrm{LT}}$. The rule ψ can be of the following forms:
  - ψ is a deductive ground rule. It must be that $\boldsymbol{f}$ is well-formed since $\mathit{pos}(\psi)$ is well-formed, and the location specifier and timestamp in the head are those of the facts in $\mathit{pos}(\psi)|_{\mathit{sch}(P)^{\mathrm{LT}}}$.

– ψ is an inductive ground rule. Like in the previous item we obtain that the location specifier in the head is in $\mathcal{N}$. In addition, the timestamp in the head is in $\mathbb{N}$ because it occurs in a tsucc-fact in $pos(\psi)$ and the only values occuring in tsucc-facts are $\mathbb{N}$. Therefore $\boldsymbol{f}$ is well-formed.

– ψ is of the form (4.11):

$$R(y, t, \bar{a}) \leftarrow R_{\text{snd}}(x, s, y, t, \bar{a}).$$

Now, $\boldsymbol{f}$ is well-formed because $R_{\text{snd}}(x, s, y, t, \bar{a})$ is well-formed.

- Suppose $\boldsymbol{f}$ is of the form $\texttt{rcvClock}(x, s, y, t)$, $\texttt{isBehind}(x, s, y, t)$ or $\texttt{clock}(x, s, y, t)$. Then ψ can be of following forms (4.3), (4.4), (4.5), (4.6), (4.7) and (4.12) respectively:

$$\texttt{rcvClock}(x, s, y, s) \leftarrow \texttt{all}(x), \texttt{all}(y), x \neq y, \texttt{zero}(s).$$

$$\texttt{rcvClock}(x, s, x, s') \leftarrow \texttt{all}(x), \texttt{tsucc}(s, s').$$

$$\texttt{rcvClock}(x, s', y, t) \leftarrow \texttt{clock}(x, s, y, t), x \neq y, \texttt{tsucc}(s, s').$$

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

$$\texttt{clock}(x, s, y, t) \leftarrow \texttt{rcvClock}(x, s, y, t).$$

$$\texttt{rcvClock}(y, t, z, u) \leftarrow R_{\text{snd}}(x, s, y, t, \bar{v}), \texttt{clock}(x, s, z, u).$$

Because $pos(\psi)$ is well-formed, it is clear that $\boldsymbol{f}$ is well-formed.

- Suppose $\boldsymbol{f}$ is of the form $R_{\text{snd}}(x, s, y, t, \bar{a})$, $\texttt{chosen}_R(x, s, y, t, \bar{a})$ or $\texttt{other}_R(x, s, y, \bar{a}, t)$ with R a relation name in $sch(P)$. This is similar to the above.
- Suppose $\boldsymbol{f}$ is of the form $\texttt{zero}(t)$ or $\texttt{notZero}(t)$. This is similar to the above.

□

D. TRACE

D.1 Vector clocks

The following properties provide insight into the vector clocks we associated with each global transition index of a run, as defined in Section 5.1. Let $\preceq$ denote the partial order on vector clocks.

Proposition D.1. *Consider Setting B.1. Let i be a global transition index. For $y \in \mathcal{N}$, if $x_i \neq y$ then $v_{\mathcal{R}}(i)[y] \leq local_{\mathcal{R}}(i, y)$.*

Proof. We show this by induction on i. For the base case ($i = 0$), we have $v_{\mathcal{R}}(i)[y] = 0$ by definition and the property trivially holds. For the induction hypothesis we assume that the property holds for all global transition indices up to and including $i - 1$ with $i - 1 \geq 0$. For the inductive step, we show that the property holds for global transition index i. We have $i > 0$, so by definition $v_{\mathcal{R}}(i)[y] = \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$. Let $u \in \{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y)$. We show that $u \leq local_{\mathcal{R}}(i, y)$. For $u = 0$ this trivially holds.

- Suppose that $u \in \mu_{\mathcal{R}}(i, y)$. By definition of $\mu_{\mathcal{R}}(i, y)$ there is a global transition $k < i$ and a fact $\boldsymbol{f} \in \delta_k$ such that $\alpha(k, \boldsymbol{f}) = i$ and $u = v_{\mathcal{R}}(k)[y]$. If $x_k = y$ then by definition $u = local_{\mathcal{R}}(k, y) + 1$, but since $k < i$ we have $local_{\mathcal{R}}(k, y) + 1 \leq local_{\mathcal{R}}(i, y)$. If $x_k \neq y$ then by the induction hypothesis $v_{\mathcal{R}}(k)[y] \leq local_{\mathcal{R}}(k, y) \leq local_{\mathcal{R}}(i, y)$. So in either case $u \leq local_{\mathcal{R}}(i, y)$.
- Suppose that $u \in \pi_{\mathcal{R}}(i, y)$. By definition of $\pi_{\mathcal{R}}(i, y)$ there is a global transition index $j \in prev_{\mathcal{R}}(i)$ such that $x_j = x_i$ and $u = v_{\mathcal{R}}(j)[y]$. Now since $x_j \neq y$ and $j < i$ we can apply the induction hypothesis to know $v_{\mathcal{R}}(j)[y] \leq local_{\mathcal{R}}(j, y) \leq local_{\mathcal{R}}(i, y)$.

□

Lemma D.2. *Consider Setting B.1. For global transition index i, and $j \in prev_{\mathcal{R}}(i)$, we have $v_{\mathcal{R}}(j) \prec v_{\mathcal{R}}(i)$.*

Proof. Denote $x = x_i$. We have $x = x_j$ and $j < i$ by definition of $prev_{\mathcal{R}}(i)$. Now, by definition $v_{\mathcal{R}}(j)[x] = local_{\mathcal{R}}(j) + 1$ and $v_{\mathcal{R}}(i)[x] = local_{\mathcal{R}}(i) + 1$. But since $j < i$ we have $local_{\mathcal{R}}(j) < local_{\mathcal{R}}(i)$ and thus $v_{\mathcal{R}}(j)[x] < v_{\mathcal{R}}(i)[x]$.

Now let $y \in \mathcal{N}$ with $y \neq x$. We show $v_{\mathcal{R}}(j)[y] \leq v_{\mathcal{R}}(i)[y]$. By definition, $v_{\mathcal{R}}(j)[y] \in \pi_{\mathcal{R}}(i, y)$ and thus by definition of $v_{\mathcal{R}}(i)[y]$ we have $v_{\mathcal{R}}(j)[y] \leq v_{\mathcal{R}}(i)[y]$.

Overall, we have $v_{\mathcal{R}}(j) \prec v_{\mathcal{R}}(i)$. □

Lemma D.3. *Consider Setting B.1. For global transition indices i and j with $j < i$ we have $v_{\mathcal{R}}(j)[x_i] < v_{\mathcal{R}}(i)[x_i]$.*

Proof. If $x_j = x_i$ then by definition of $v_{\mathcal{R}}(j)$ and $v_{\mathcal{R}}(i)$ we have $v_{\mathcal{R}}(j)[x_i] = local_{\mathcal{R}}(j) + 1 < local_{\mathcal{R}}(i) + 1 = v_{\mathcal{R}}(i)[x_i]$.

Now suppose $x_j \neq x_i$. By Proposition D.1 we then have $v_{\mathcal{R}}(j)[x_i] \leq local_{\mathcal{R}}(j, x_i)$. Since $j < i$ we have $local_{\mathcal{R}}(j, x_i) \leq local_{\mathcal{R}}(i, x_i)$. By definition, $v_{\mathcal{R}}(i)[x_i] = local_{\mathcal{R}}(i, x_i) + 1$ and thus $v_{\mathcal{R}}(j)[x_i] < v_{\mathcal{R}}(i)[x_i]$. □

D.2 Trace definition

Consider the following lemma:

Lemma D.4. *Consider Setting B.1. Let i be a global transition of $\mathcal{R}$. Let $\texttt{rcvClock}(y, t, z, u)$ be a fact in the set (5.9) for i. If $y = x_i$ then $t \geq local_{\mathcal{R}}(i) + 1$.*

Proof. By definition of the set (5.9) for i, there is a fact $\boldsymbol{f} \in \delta_i$ such that $t = local_{\mathcal{R}}(\alpha(i, \boldsymbol{f}))$. By definition of α this implies $i < \alpha(i, \boldsymbol{f})$. Denote $j = \alpha(i, \boldsymbol{f})$. By definition of y, we have $y = x_j$. By assumption $y = x_i$ and thus $x_j = x_i$. Then $local_{\mathcal{R}}(i) < local_{\mathcal{R}}(j)$ because $i < j$, and thus $local_{\mathcal{R}}(i) < t$. We obtain that $local_{\mathcal{R}}(i) + 1 \leq t$. □

Lemma D.5. *Consider Setting B.1. Let M be the trace of $\mathcal{R}$. We have $decl(H) \subseteq M$.*

Proof. Recall that by definition $decl(H)$ consists of the following facts:

1. $\bigcup_{x \in \mathcal{N}} \bigcup_{s \in \mathbb{N}} (H|^x)^{\uparrow s}$;
2. $decl(\mathcal{N})$;
3. I_{time}.

First, by definition of M_{-1} we have $decl(\mathcal{N}) \subseteq M_{-1}$ and $I_{\text{time}} \subseteq M_{-1}$. Since $M_{-1} \subseteq M$ we thus have $decl(\mathcal{N}) \subseteq M$ and $I_{\text{time}} \subseteq M$. In order to obtain $decl(H) \subseteq M$, now we check that M also includes the set $\bigcup_{x \in \mathcal{N}} \bigcup_{s \in \mathbb{N}} (H|^x)^{\uparrow s}$:

$$\begin{aligned}
\bigcup_{x \in \mathcal{N}} \bigcup_{s \in \mathbb{N}} (H|^x)^{\uparrow s} &= \bigcup_{j \in \mathbb{N}} (H|^{x_j})^{\uparrow local_{\mathcal{R}}(j)} && (*) \\
&\subseteq \bigcup_{j \in \mathbb{N}} (s^{\rho_j}|^{x_j})^{\uparrow local_{\mathcal{R}}(j)} && (**) \\
&\subseteq \bigcup_{j \in \mathbb{N}} S_j^{\uparrow local_{\mathcal{R}}(j)} \\
&\subseteq \bigcup_{j \in \mathbb{N}} D_j^{\uparrow local_{\mathcal{R}}(j)} \\
&\subseteq \bigcup_{j \in \mathbb{N}} M_j \\
&\subseteq M,
\end{aligned}$$

where $(*)$ follows from Lemma B.3 and $(**)$ from Lemma B.2, and the other inclusions follow from the definitions of S_j, D_j and the set (5.1) for j. □

E. RUN TO STABLE MODEL

In this section we show that the trace of a (fair) run is a fair stable model of the declarative semantics. All proofs will be about the following setting:

Setting E.1. Let P be a Dedalus program. Let H be an input distributed database instance, over a network $\mathcal{N}$. Let $\mathcal{R}$ be a run of P on input H. Let α be the arrival function for $\mathcal{R}$. Let M be the trace of $\mathcal{R}$, as defined in Section 5, so M is a union of sets $\bigcup_{i \in \mathbb{N} \cup \{-1\}} M_i$ as defined there. Let N be the output of $G_M(P)$ on input $decl(H)$. □

We reuse the notations from the previous sections.

E.1 Properties

Lemma E.2. *Consider Setting E.1. We have $M \subseteq N$.*

Proof. By definition $M = \bigcup_{i \in \mathbb{N} \cup \{-1\}} M_i$. We show by induction that for $i \in \mathbb{N} \cup \{-1\}$ we have $M_i \subseteq N$. For the base case ($i = -1$), the property holds by Lemma E.3. For the induction hypothesis we assume that $M_{i-1} \subseteq N$ with $i - 1 \geq -1$. This implies $M_j \subseteq N$ for $j = -1, 0, \ldots, i-1$. For the inductive step, we show that $M_i \subseteq N$. Denote $s = local_{\mathcal{R}}(i)$. We have $s \geq 0$ by definition of $local_{\mathcal{R}}(i)$.

State (5.1).

For set (5.1), we show that $D_i^{\uparrow s} \subseteq N$. Abbreviate $S_i = s^{\rho_i}|^{x_i} \cup \mathit{untag}(m_i)$.

We first show that $S_i^{\uparrow s} \subseteq N|^{x_i,s}$ and we start with showing that $(s^{\rho_i}|^{x_i})^{\uparrow s} \subseteq N|^{x_i,s}$:

1. If $s = 0$ then $\rho_0 = \mathit{start}(P, H)$ and thus $s^{\rho_i}|^{x_i} = H^{x_i}$ by definition of $\mathit{start}(P, H)$. And in that case, $(s^{\rho_i}|^{x_i})^{\uparrow s} \subseteq N^{\blacktriangle}|^{x_i,s} \subseteq N|^{x_i,s}$ because $(H|^{x_i})^{\uparrow s} \subseteq \mathit{decl}(H) \subseteq N$ and by definition of $N^{\blacktriangle}$.
2. If $s > 0$, then $\mathit{prev}_{\mathcal{R}}(i) \neq \emptyset$. Let $j \in \mathit{prev}_{\mathcal{R}}(i)$. By definition of $\mathit{prev}_{\mathcal{R}}(i)$ we have $j < i$. By the operational semantics of the run $\mathcal{R}$, we have $(s^{\rho_i}|^{x_i})^{\uparrow s} = I_j^{\uparrow s}$, with $I_j = \mathit{induc}(P)(D_j)$. We have $D_j^{\uparrow s-1} \subseteq M_j$ by definition of the set (5.1) for j. By applying the induction hypothesis to j we then have $D_j^{\uparrow s-1} \subseteq N$. Now by applying Lemma E.5 we have $I_j^{\uparrow s} \subseteq N|^{x_i,s}$.

Now we show that $\mathit{untag}(m_i)^{\uparrow s} \subseteq N|^{x_i,s}$. Consider $R(x_i, \bar{a}) \in \mathit{untag}(m_i)$. Using the operational semantics, there must exist a previous global transition $k < i$ such that $R(x_i, \bar{a}) \in \delta_k$, i.e., the global transition where $R(x_i, \bar{a})$ is sent. By definition of the set (5.6) for k we have $R_{\mathrm{snd}}(x_k, l, x_i, s, \bar{a}) \in M_k$ where $l = \mathit{local}_{\mathcal{R}}(k)$. Now, by applying the induction hypothesis we have $M_k \subseteq N$ and thus the following ground rule of $G_M(P)$, based on the form (4.11), derives $R(x_i, s, \bar{a}) \in N|^{x_i,s}$:

$$R(x_i, s, \bar{a}) \leftarrow R_{\mathrm{snd}}(x_k, l, x_i, s, \bar{a}).$$

We obtain that overall $S_i^{\uparrow s} \subseteq N|^{x_i,s}$. Now we apply Lemma E.4 to obtain that $D_i^{\uparrow s} \subseteq N|^{x_i,s} \subseteq N$.

Clock (5.2).

For the set (5.2) of global transition i, we show that $\texttt{rcvClock}(x_i, s, x_i, s+1) \in N$. We have the following ground rule in $G_M(P)$, based on the form (4.4):

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

Therefore $\texttt{rcvClock}(x_i, s, x_i, s+1) \in N$ because N is a fixpoint.

Clock (5.3).

Let $y \in \mathcal{N}$ with $x_i \neq y$ and let $t \in \pi_{\mathcal{R}}(i, y)$. We show that $\texttt{rcvClock}(x_i, s, y, t) \in N$.

Because $\pi_{\mathcal{R}}(i, y) \neq \emptyset$, we have $\mathit{prev}_{\mathcal{R}}(i) \neq \emptyset$ and $s > 0$. Let $j \in \mathit{prev}_{\mathcal{R}}(i)$. We have $j < i$. We have $\mathit{local}_{\mathcal{R}}(j) = \mathit{local}_{\mathcal{R}}(i) - 1 = s - 1$. By definition of the set (5.5) for j, set M_j contains $\texttt{clock}(x_i, s-1, y, t)$ with $t = v_{\mathcal{R}}(j)[y]$. Therefore $\texttt{clock}(x_i, s-1, y, t) \in N$ by applying the induction hypothesis to M_j. In $G_M(P)$ we have the following ground rule, based on the form (4.5):

$$\texttt{rcvClock}(x_i, s, y, t) \leftarrow \texttt{clock}(x_i, s-1, y, t), x_i \neq y, \texttt{tsucc}(s-1, s).$$

This implies $\texttt{rcvClock}(x_i, s, y, t) \in N$ because N is a fixpoint.

Clock (5.4).

We take two facts $\texttt{rcvClock}(x_i, s, y, t)$ and $\texttt{rcvClock}(x_i, s, y, t')$ in M_i with $t < t'$. We show that $\texttt{isBehind}(x_i, s, y, t) \in N$.

First we show that $\texttt{rcvClock}(x_i, s, y, t) \in N$ and $\texttt{rcvClock}(x_i, s, y, t') \in N$. By Lemma D.4, there are only three kinds of $\texttt{rcvClock}$-facts in M_i with first two components x_i and s: *(i)* $\texttt{rcvClock}$-facts in M_{i-1}, *(ii)* facts in the set (5.2) for i, and *(iii)* facts in the set (5.3) for i. We can apply the induction hypothesis to know that the first kind is in N. Also, we have shown above that the second and third kinds are also in N.

Now consider the following ground rule, based on the form (4.6):

$$\begin{aligned} \texttt{isBehind}(x_i, s, y, t) \quad \leftarrow \quad & \texttt{rcvClock}(x_i, s, y, t), \\ & \texttt{rcvClock}(x_i, s, y, t'), t < t'. \end{aligned}$$

Because $\texttt{rcvClock}(x_i, s, y, t) \in N$, $\texttt{rcvClock}(x_i, s, y, t') \in N$ and N is a fixpoint, we have $\texttt{isBehind}(x_i, s, y, t) \in N$.

Clock (5.5).

Let $y \in \mathcal{N}$. Denote $t = v_{\mathcal{R}}(i)[y]$. We show that $\texttt{clock}(x_i s, y, t) \in N$.

Suppose we would already know that $\texttt{rcvClock}(x_i, s, y, t) \in N$ and $\texttt{isBehind}(x_i, s, y, t) \notin M$. Then, based on the form (4.7), the following ground rule is in $G_M(P)$:

$$\texttt{clock}(x_i, s, y, t) \leftarrow \texttt{rcvClock}(x_i, s, y, t).$$

And thus $\texttt{clock}(x_i, s, y, t) \in N$ because N is a fixpoint.

Now, to actually show that $\texttt{rcvClock}(x_i, s, y, t) \in N$ and $\texttt{isBehind}(x_i, s, y, t) \notin M$ we consider two cases: $y = x_i$ and $y \neq x_i$. To show that $\texttt{isBehind}(x_i, s, y, t) \notin M$, it is sufficient to show that $\texttt{isBehind}(x_i, s, y, t) \notin M_i$, because by Lemma B.3, the set (5.4) of global transition i is the only part of M where we add $\texttt{isBehind}$-facts with first two components x_i and s.

First, suppose $y = x_i$. We show that $\texttt{clock}(x_i, s, x_i, t) \in N$. By definition $v_{\mathcal{R}}(i)[x_i] = \mathit{local}_{\mathcal{R}}(i) + 1 = s + 1$. We have shown above that $\texttt{rcvClock}(x_i, s, x_i, s+1) \in N$, by set (5.2) for i. Now we show that $\texttt{isBehind}(x_i, s, x_i, s+1) \notin M_i$. Let $\texttt{rcvClock}(x_i, s, x_i, u) \in M_i$ with $u \neq s+1$. We abbreviate this fact as $\boldsymbol{f}$. We show that $u < s+1$. If $\boldsymbol{f} \in M_{-1}$ then $u = 0$, and the inequality holds immediately. Now suppose $\boldsymbol{f} \notin M_{-1}$. There are several possible cases:

1. The fact $\boldsymbol{f}$ cannot be generated by the set (5.2) for i because that requires $u = s + 1$.

2. The fact $\boldsymbol{f}$ cannot be generated by set (5.3) for i because that requires the first and third component to be different.
3. The fact $\boldsymbol{f}$ cannot be generated by the set (5.9) for i because facts in this set with first component x_i have a second component that is at least $s+1$ by Lemma D.4.
4. The last option is that there is a global transition index $j < i$ with $\texttt{rcvClock}(x_i, s, x_i, u) \in M_j \setminus M_{j-1}$. Denote $s' = \mathit{local}_{\mathcal{R}}(j)$. The fact $\boldsymbol{f}$ cannot be in the sets (5.2) or (5.3) for j because $s' < s$. So, the final possibility is that the set (5.9) for j contains $\boldsymbol{f}$. This implies $u = v_{\mathcal{R}}(j)[x_i]$. Now by applying Lemma D.3 we obtain that $v_{\mathcal{R}}(j)[x_i] < v_{\mathcal{R}}(i)[x_i] = s+1$.

So, there is no value $u \in \mathbb{N}$ with $\texttt{rcvClock}(x_i, s, x_i, u) \in M_i$ and $s+1 < u$. Then, by construction of M_i we have not added the fact $\texttt{isBehind}(x_i, s, x_i, s+1)$ to the set (5.4) for i.

Now suppose $y \neq x_i$. We first show that $\texttt{clock}(x_i, s, y, t) \in N$. Denote $t = v_{\mathcal{R}}(i)[y]$. By definition, $t = \max(\{0\} \cup \mu_{\mathcal{R}}(i,y) \cup \pi_{\mathcal{R}}(i,y))$ where $\mu_{\mathcal{R}}(i,y)$ and $\pi_{\mathcal{R}}(i,y)$ are as defined in Section 5.1. If both $\mu_{\mathcal{R}}(i,y) = \emptyset$ and $\pi_{\mathcal{R}}(i,y) = \emptyset$ then $s = 0$ and by definition $t = 0$, but by Lemma E.3 we have $\texttt{rcvClock}(x_i, 0, y, 0) \in N$. Now suppose that at least one of $\mu_{\mathcal{R}}(i,y)$ and $\pi_{\mathcal{R}}(i,y)$ is not empty:

- Let $u \in \mu_{\mathcal{R}}(i,y)$. We show that $\texttt{rcvClock}(x_i, s, y, u) \in N$. By definition of $\mu_{\mathcal{R}}(i,y)$ there is a global transition $k < i$ and a fact $\boldsymbol{f} \in \delta_k$ with $\alpha(k, \boldsymbol{f}) = i$ and $v_{\mathcal{R}}(k)[y] = u$. The fact $\boldsymbol{f}$ is of the form $R(x_i, \bar{a})$. Now by the set (5.9) for k, we have $\texttt{rcvClock}(x_i, s, y, u) \in M_k$. By applying the induction hypothesis to M_k we then have $\texttt{rcvClock}(x_i, s, y, u) \in N$.
- Let $u \in \pi_{\mathcal{R}}(i,y)$. We show that $\texttt{rcvClock}(x_i, s, y, u) \in N$. By definition of $\pi_{\mathcal{R}}(i,y)$, there exists $j \in \mathit{prev}_{\mathcal{R}}(i)$ such that $v_{\mathcal{R}}(j)[y] = u$. By definition of $\mathit{prev}_{\mathcal{R}}(i)$, this implies $s > 0$, $x_j = x_i$ and $\mathit{local}_{\mathcal{R}}(j) = s-1$. By construction of M_j, the set (5.5) for j contains $\texttt{clock}(x_j, s-1, y, u)$. Thus, by substituting $x_j = x_i$ and by applying the induction hypothesis to M_j, we obtain $\texttt{clock}(x_i, s-1, y, u) \in N$. Since $y \neq x_i$ the following ground rule, based on (4.5), exists in $G_M(P)$ and it derives $\texttt{rcvClock}(x_i, s, y, u) \in N$:

$$\begin{aligned}\texttt{rcvClock}(x_i, s, y, u) \quad \leftarrow \quad & \texttt{clock}(x_i, s-1, y, u), x_i \neq y, \\ & \texttt{tsucc}(s-1, s).\end{aligned}$$

So, overall, we have $\texttt{rcvClock}(x_i, s, y, t) \in N$. Now we show that $\texttt{isBehind}(x_i, s, y, t) \notin M_i$. Let $\texttt{rcvClock}(x_i, s, y, u) \in M_i$ with $u \neq t$. We abbreviate this fact as $\boldsymbol{f}$. We show that $u < t$. By construction of M and by definition of $v_{\mathcal{R}}(i)[y]$ we have $u \geq 0$ and $t \geq 0$ respectively. So, if $u = 0$ then $t > 0$, and the inequality immediately holds. Now suppose $u > 0$, which implies $\boldsymbol{f} \notin M_{-1}$. There are several options:

1. The fact $\boldsymbol{f}$ can not be in the set (5.2) for i because it requires $x_i = y$.
2. It is possible that $\boldsymbol{f}$ is in the set (5.3) for i and in that case $u = v_{\mathcal{R}}(j)[y]$ with $j \in \mathit{prev}_{\mathcal{R}}(i)$. But then $u \in \pi_{\mathcal{R}}(i,y)$ and thus $u \leq t$. Since by assumption $u \neq t$, we have $u < t$.
3. The fact $\boldsymbol{f}$ can not be in the set (5.9) for i because if those facts have first component x_i, then their second component is at least $s+1$ by Lemma D.4.
4. It is possible that there is a global transition $j < i$ with $\boldsymbol{f} \in M_j \setminus M_{j-1}$. Again, $\boldsymbol{f}$ can not be in the set (5.2) for j because that requires $x_i = y$. If $x_j = x_i$, then $\boldsymbol{f}$ can not be in the set (5.3) for j, because $\mathit{local}_{\mathcal{R}}(j) < \mathit{local}_{\mathcal{R}}(i)$. The final option is that $\boldsymbol{f}$ is in the set (5.9) for j. This implies $u = v_{\mathcal{R}}(j)[y]$. By definition of the set (5.9) for j, there is a fact $\boldsymbol{g} \in \delta_j$ of the form $R(x_i, \bar{a})$ with $s = \mathit{local}_{\mathcal{R}}(\alpha(j, \boldsymbol{g}))$ where x_i is the recipient during global transition $\alpha(j, \boldsymbol{g})$. Denote $h = \alpha(j, \boldsymbol{g})$. Because x_i is the recipient during global transitions i and h and because $\mathit{local}_{\mathcal{R}}(i) = s = \mathit{local}_{\mathcal{R}}(h)$ we have $i = h$ by Lemma B.3. Thus by definition of $\mu_{\mathcal{R}}(i,y)$ we have $u = v_{\mathcal{R}}(j)[y] \in \mu_{\mathcal{R}}(i,y)$ and therefore $u \leq t$ by definition of t. But since $u \neq t$ we have $u < t$.

Thus there is no value $u \in \mathbb{N}$ with $\texttt{rcvClock}(x_i, s, y, u) \in M_i$ and $t < u$. By definition of the set (5.4) for i, we obtain that $\texttt{isBehind}(x_i, s, y, t) \notin M_i$.

Message sending **(5.6), (5.7), (5.8), (5.9).**

Let $\boldsymbol{f} \in \delta_i$, which is a fact of the form $R(y, \bar{a})$. Denote $k = \alpha(i, \boldsymbol{f})$, $t = \mathit{local}_{\mathcal{R}}(k)$ and $u = v_{\mathcal{R}}(i)[y]$.

We first show that $u \leq t$. First, by definition of α, we have $i < k$. Because y is the recipient during transition k, by Lemma D.3 we have $u < v_{\mathcal{R}}(k)[y]$. By definition, $v_{\mathcal{R}}(k)[y] = t+1$. Therefore $u < t+1$ and thus $u \leq t$.

Let φ and V be an asynchronous (simplified) rule of $\mathit{async}(P)$ and a valuation respectively that together produced $\boldsymbol{f} \in \delta_i$ on input D_i. We will use φ and V to show inclusion of the sets (5.6), (5.7), (5.8) and (5.9) in N. But before we continue we need to map φ to rules in $\mathit{pure}(P)$. For this purpose, let φ_2 be an original, unsimplified rule of P such that φ is the simplified version of φ_2. The rule φ_2 is of the following form:

$$\begin{aligned}R(y', t', \bar{v}) \quad \leftarrow \quad & \mathbf{B}\{x', s' \mid \bar{v}, \bar{w}, y'\}, \texttt{time}(t'), \\ & \mathit{choice}(\langle x', s', y', \bar{v}\rangle, \langle t'\rangle);\end{aligned}$$

where we have added a prime ($'$) to some variables to distinguish them from some symbols we use here for *values*. Let us denote the R_{snd}-rule in $\mathit{pure}(P)$ that is based on φ_2 (and the form (4.8)) as follows:

$$\begin{aligned}R_{\mathrm{snd}}(x', s', y', t', \bar{v}) \quad \leftarrow \quad & \mathbf{B}\{x', s' \mid \bar{v}, \bar{w}, y'\}, \texttt{all}(y'), \\ & \texttt{clock}(x', s', y', u'), \texttt{time}(t'), u' \leq t', \\ & \texttt{chosen}_R(x', s', y', \bar{v}, t').\end{aligned}$$

Let V_2 be V extended with the additional mappings $s' \mapsto s$, $t' \mapsto t$ and $u' \mapsto u$. Because V is satifying for φ, we have $V(pos(\varphi)) \subseteq D_i$ and thus $V_2(pos(\varphi_2)|_{sch(P)}) \subseteq D_i^{\uparrow s}$ by definition of φ from φ_2. And as we have shown above for the set 5.1 for i that $D_i^{\uparrow s} \subseteq N$ and thus $V_2(pos(\varphi_2)|_{sch(P)}) \subseteq N$. Also, $V(neg(\varphi)) \cap D_i = \emptyset$ and thus $V_2(neg(\varphi_2)) \cap D_i^{\uparrow s} = \emptyset$ (using that $neg(\varphi_2)|_{sch(P)} = neg(\varphi_2)$). Specifically, the facts in $V_2(neg(\varphi_2))$ are over the schema $sch(P)^{\mathrm{LT}}$ and have location specifier x_i and timestamp s. By Lemma B.3, the set (5.1) for i is the only part of M where we add facts over the schema $sch(P)^{\mathrm{LT}}$ with location specifier x_i and timestamp s. Therefore $V_2(neg(\varphi_2)) \cap M = \emptyset$.

Now, for the set (5.7) of global transition i, we can show that $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$. By construction of the sets (5.7) and (5.8) for i, we have added to M_i the facts $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t)$ and $\mathtt{other}(x_i, s, y, \bar{a}, t_2)$ with $t_2 \geq u$ and $t_2 \neq t$. The absence of the facts $\mathtt{other}(x_i, s, y, \bar{a}, t)$ and $V_2(neg(\varphi_2))$ from M imply that the following positive ground rule is present in $G_M(P)$, based on the above R_{snd}-rule by applying valuation V_2 to the variables and respecting the form (4.9):

$$\begin{aligned}\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \quad \leftarrow \quad & V_2(pos(\varphi_2)|_{sch(P)}), V_2(neq(\varphi_2)), \mathtt{all}(y),\\ & \mathtt{clock}(x_i, s, y, u), \mathtt{time}(t), u \leq t.\end{aligned}$$

We now show that the body of this ground rule is true on N. From above we have $V_2(pos(\varphi_2)|_{sch(P)}) \subseteq N$ and the ground nonequalities $V_2(neq(\varphi_2))$ must be satisfied because V is satisfying for φ. We have shown above for the set (5.5) for i that $\mathtt{clock}(x_i, s, y, u) \in N$. We have $\mathtt{all}(y) \in decl(H)$ and $\mathtt{time}(t) \in decl(H)$ and thus $\mathtt{all}(y) \in N$ and $\mathtt{time}(t) \in N$ by definition of N. Also, we have shown above that $u \leq t$. Therefore, overall, the body of the above ground $\mathtt{chosen}_R$-rule is true on N and thus $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$ because N is a fixpoint.

Now let $t_2 \in \mathbb{N}$ with $v_{\mathcal{R}}(i)[y] \leq t_2$ and $t_2 \neq t$. Thus $u \leq t_2$, by definition of u. For the set (5.8) of global transition i, we show that $\mathtt{other}_R(x_i, s, y, \bar{a}, t_2) \in N$. Reusing the rule φ_2 and valuation V_2 from above, we have the following ground rule in $G_M(P)$, based on the form (4.10):

$$\begin{aligned}\mathtt{other}_R(x_i, s, y, \bar{a}, t_2) \quad \leftarrow \quad & V_2(pos(\varphi_2)|_{sch(P)}), V_2(neq(\varphi_2)), \mathtt{all}(y),\\ & \mathtt{clock}(x_i, s, y, u), \mathtt{time}(t_2), u \leq t_2,\\ & \mathtt{chosen}_R(x_i, s, y, \bar{a}, t), t_2 \neq t.\end{aligned}$$

Since $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$ we have $\mathtt{other}_R(x_i, s, y, \bar{a}, t') \in N$ because N is a fixpoint.

For the set (5.6) of global transition i, we show that $R_{\mathrm{snd}}(x_i, s, y, t, \bar{a}) \in N$. We have the following ground rule, based on the above R_{snd}-rule:

$$\begin{aligned}R_{\mathrm{snd}}(x_i, s, y, t, \bar{a}) \quad \leftarrow \quad & V_2(pos(\varphi_2)|_{sch(P)}), V_2(neq(\varphi_2)), \mathtt{all}(y),\\ & \mathtt{clock}(x_i, s, y, u), \mathtt{time}(t), u \leq t,\\ & \mathtt{chosen}_R(x_i, s, y, \bar{a}, t).\end{aligned}$$

Since $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \in N$ we have $R_{\mathrm{snd}}(x_i, s, y, t, \bar{a}) \in N$ because N is a fixpoint.

Let $z \in \mathcal{N}$. Denote $v = v_{\mathcal{R}}(i)[z]$. For the set (5.9) of global transition i, we show that $\mathtt{rcvClock}(y, t, z, v) \in N$. The following ground rule is in $G_M(P)$, based on the rule (4.12), because the rule 4.12 contains no negation:

$$\mathtt{rcvClock}(y, t, z, v) \leftarrow R_{\mathrm{snd}}(x_i, s, y, t, \bar{a}), \mathtt{clock}(x_i, s, z, v).$$

First, we have already shown for the set (5.5) of i that $\mathtt{clock}(x_i, s, z, v) \in N$. Now since $R_{\mathrm{snd}}(x_i, s, y, t, \bar{a}) \in N$ from above, we have $\mathtt{rcvClock}(y, t, z, v) \in N$ because N is a fixpoint. ☐

Lemma E.3. *Consider Setting E.1. We have $M_{-1} \subseteq N$.*

Proof. Recall that M_{-1} consists of the following facts:

1. $decl(\mathcal{N})$ and I_{time};
2. $\{\mathtt{rcvClock}(x, 0, y, 0) \mid x, y \in \mathcal{N}\}$;
3. $\{\mathtt{notZero}(j) \mid j \in \mathbb{N} \setminus \{0\}\}$;
4. $\mathtt{zero}(0)$.

For each group of facts we show inclusion in N:

1. We have $decl(\mathcal{N}) \cup I_{\mathrm{time}} \subseteq decl(H)$ by definition of $decl(H)$. We have $decl(H) \subseteq N$ by definition of N.
2. These facts are created by the ground rules in $G_M(P)$ of the form (4.3):
$$\mathtt{rcvClock}(x, s, y, s) \leftarrow \mathtt{all}(x), \mathtt{all}(y), x \neq y, \mathtt{zero}(s).$$
3. For $j \in \mathbb{N} \setminus \{0\}$ the following ground rule derives $\mathtt{notZero}(j)$, based on the form (4.1):
$$\mathtt{notZero}(j) \leftarrow \mathtt{tsucc}(j-1, j).$$
4. By construction of M, there is no fact $\mathtt{notZero}(0) \in M$. Therefore, the following ground rule, based on the form (4.2), exists in $G_M(P)$ and it derives $\mathtt{zero}(0)$:
$$\mathtt{zero}(0) \leftarrow \mathtt{time}(0).$$

□

Lemma E.4. *Consider Setting E.1. Let i be a global transition of $\mathcal{R}$. Denote $s = local_{\mathcal{R}}(i)$. Suppose that $S_i^{\uparrow s} \subseteq N|^{x_i,s}$. We have $D_i^{\uparrow s} \subseteq N|^{x_i,s}$.*

Proof. The reasoning of this proof is similar to the known reasoning used to show that stratified semantics and stable model semantics coincide for syntactically stratified programs.

Recall that by definition $D_i = \mathit{deduc}(P)(S_i)$.

Abbreviate $A_k = \mathit{deduc}_k(P)(S_i)$. We show by induction on the stratum numbers k that $A_k^{\uparrow s} \subseteq N|^{x_i,s}$. For the base case ($k = 0$), we have $A_0 = S_i$. We are given that $S_i^{\uparrow s} \subseteq N|^{x_i,s}$ and thus $A_0^{\uparrow s} \subseteq N|^{x_i,s}$. For the induction hypothesis, assume that the property holds for stratum $k-1$ with $k-1 \geq 0$: $A_{k-1}^{\uparrow s} \subseteq N|^{x_i,s}$. For the inductive step, we show that $A_k^{\uparrow s} \subseteq N|^{x_i,s}$. Since $A_{k-1} \subseteq A_k$, it is sufficient to show that $(A_k \setminus A_{k-1})^{\uparrow s} \subseteq N|^{x_i,s}$. We can consider the computation of stratum k in A_k to be a fixpoint computation. So, we have a sequence of fact-sets $A_k^0 \subseteq A_k^1 \subseteq A_k^2 \subseteq \ldots$ with $A_k^0 = A_{k-1}$. We now show by induction on $j = 0, 1, 2, \ldots$ that $(A_k^j)^{\uparrow s} \subseteq N|^{x_i,s}$:

- For the base case ($j = 0$) we have $(A_k^0)^{\uparrow s} = (A_{k-1})^{\uparrow s} \subseteq N|^{x_i,s}$ by applying the outer induction hypothesis.
- For the induction hypothesis, assume that $(A_k^{j-1})^{\uparrow s} \subseteq N|^{x_i,s}$ with $j-1 \geq 0$.
- For the inductive step, we show that $(A_k^j)^{\uparrow s} \subseteq N|^{x_i,s}$. Let $\boldsymbol{f} \in A_k^j \setminus A_k^{j-1}$. Let φ and V be the simplified deductive rule in $\mathit{deduc}_k(P)$ and valuation that have produced $\boldsymbol{f}$ on input A_k^{j-1}. Let φ_2 be the original, unsimplified rule of P on which φ is based. Let V_2 be V extended to assign the value s to the body timestamp variable of φ_2. We have $V(\mathit{pos}(\varphi)) \subseteq A_k^{j-1}$ and thus $V_2(\mathit{pos}(\varphi_2)) \subseteq (A_k^{j-1})^{\uparrow s} \subseteq N|^{x_i,s}$ by applying the inner induction hypothesis. Also, $V(\mathit{neg}(\varphi)) \cap A_{k-1} = \emptyset$ since negation in a stratified program can only be applied to lower strata. Because a relation only belongs to one stratum of $\mathit{deduc}(P)$, we additionally have $V(\mathit{neg}(\varphi)) \cap D_i = \emptyset$ and thus $V_2(\mathit{neg}(\varphi_2)) \cap D_i^{\uparrow s} = \emptyset$. Now, since only in the definition of the set (5.1) for M_i do we add facts over schema $\mathit{sch}(P)^{\mathrm{LT}}$ with location specifier x_i and timestamp s, and those facts are precisely $D_i^{\uparrow s}$, we have $V_2(\mathit{neg}(\varphi_2)) \cap M = \emptyset$. Lastly, the nonequalities of φ and thus of φ_2 must be satisfied under V. Thus, the ground rule based on φ_2 and V_2 (without negative body literals) is in $G_M(P)$ and it derives the fact $V_2(\mathit{head}(\varphi_2)) = \boldsymbol{f}^{\uparrow s} \in N$ because N is a fixpoint.

To conclude, let l be the largest stratum number of $\mathit{deduc}(P)$. We have

$$\begin{aligned} D_i^{\uparrow s} &= \mathit{deduc}(P)(S_i)^{\uparrow s} \\ &= \mathit{deduc}_l(P)(S_i)^{\uparrow s} \\ &= A_l^{\uparrow s} \\ &\subseteq N|^{x_i,s}. \end{aligned}$$

□

Lemma E.5. *Consider Setting E.1. Let i be a global transition of $\mathcal{R}$. Denote $s = local_{\mathcal{R}}(i)$. Suppose that $D_i^{\uparrow s} \subseteq N|^{x_i,s}$. We have $I_i^{\uparrow s+1} \subseteq N|^{x_i,s+1}$.*

Proof. Let $\boldsymbol{f} \in I_i$. Let φ and V be the (simplified) rule of $\mathit{induc}(P)$ and valuation that have generated $\boldsymbol{f}$ on input D_i. Let φ_2 be the original, unsimplified inductive rule of P on which φ is based. The rule φ_2 has an additional positive body atom with predicate `tsucc`. Let V_2 be V extended to assign the timestamp s to the body timestamp-variable of φ_2 and to assign $s+1$ to the head timestamp-variable. We have $\texttt{tsucc}(s, s+1) \in \mathit{decl}(H) \subseteq N$. We have $V(\mathit{pos}(\varphi)) \subseteq D_i$ and using the given, $D_i^{\uparrow s} \subseteq N|^{x_i,s}$, and $\texttt{tsucc}(s, s+1) \in N$, we have $V_2(\mathit{pos}(\varphi_2)) \subseteq N$. We must also have $V(\mathit{neg}(\varphi)) \cap D_i = \emptyset$. The facts in D_i have location specifier x_i and timestamp s. Since the set (5.1) for global transition i is the only part in the definition of M where we add facts over schema $\mathit{sch}(P)^{\mathrm{LT}}$ with location specifier x_i and timestamp s, and because those facts are precisely $D_i^{\uparrow s}$, we have $V_2(\mathit{neg}(\varphi_2)) \cap M = \emptyset$. Lastly the nonequalities of φ and thus of φ_2 must be satisfied under V_2. Therefore the inductive ground rule based on φ_2 and V_2 (without negative body literals) occurs in $G_M(P)$ and it derives $V_2(\mathit{head}(\varphi_2)) = \boldsymbol{f}^{\uparrow s+1} \in N|^{x_i,s+1}$. □

Lemma E.6. *Consider Setting E.1. We have $N \subseteq M$.*

Proof. We can imagine that the facts of N are obtained by an infinite sequence of fact derivations, one fact at the time. This gives a sequence $N_0 \subseteq N_1 \subseteq N_2 \subseteq \ldots$ of sets with $N_0 = \mathit{decl}(H)$ and $N_\infty = N$. We can apply Lemma C.1 to find that N, and each of its subsets is well-formed as well. We show by induction for $i \in \mathbb{N}$ that $N_i \subseteq M$.

First we look at the base case ($i = 0$): we must show $N_0 \subseteq M$. Since $N_0 = \mathit{decl}(H)$, we can apply Lemma D.5 to know that $N_0 \subseteq M$. For the induction hypothesis, we assume that $N_{i-1} \subseteq M$ with $i-1 \geq 0$. For the inductive step we show that $N_i \subseteq M$. Let $\boldsymbol{f}$ denote the fact added to N_{i-1} in order to obtain N_i. Below, we will consider the different kinds of predicates that $\boldsymbol{f}$ can have. For each case we then show that $\boldsymbol{f} \in M$.

Facts over $sch(P)^{\mathrm{LT}}$.

Suppose that $\boldsymbol{f}$ is over a relation of $sch(P)^{\mathrm{LT}}$. Let x and s be the location specifier and timestamp of $\boldsymbol{f}$ respectively. Let ψ be the ground rule of $G_M(P)$ that derived $\boldsymbol{f}$ in N_i. This ground rule can have several forms:

- Suppose that ψ is deductive. Because ψ exists in $G_M(P)$, we can choose an original deductive rule φ of P and a valuation for φ such that $V(head(\varphi)) = head(\psi)$, $V(pos(\varphi)) = pos(\psi)$, $V(neq(\varphi)) = neq(\psi)$ and $V(neg(\varphi)) \cap M = \emptyset$. Let φ' denote the simplication of φ and let V' denote the accompanying simplification of V. Because $\mathcal{R}$ is fair and $\boldsymbol{f}$ is well-formed, by Lemma B.3 we can consider the global transition k so that its recipient x_k is x and $local_{\mathcal{R}}(k) = s$. We show that V' is satisfying for φ' during global transition k, so that $\boldsymbol{f}^{\downarrow} \in D_k$ is derived.

  We first show that $V'(pos(\varphi')) \subseteq D_k$. Because ψ derives $\boldsymbol{f}$, we have $pos(\psi) \subseteq N_{i-1} \subseteq M$ by using the induction hypothesis. Because ψ is deductive, the facts in $pos(\psi)$ are over the schema $sch(P)^{\mathrm{LT}}$ and have location specifier x and timestamp s. By Lemma B.3, it follows that the set (5.1) for global transition k is the only part of M that contains facts over schema $sch(P)^{\mathrm{LT}}$ with location specifier x and timestamp s. Thus $pos(\psi) \subseteq D_k^{\uparrow s}$. Now since by definition of φ' we have $V'(pos(\varphi'))^{\uparrow s} = V(pos(\varphi)) = pos(\psi)$, we obtain $V'(pos(\varphi'))^{\uparrow s} \subseteq D_k^{\uparrow s}$ and thus $V'(pos(\varphi')) \subseteq D_k$.

  We now show that $V'(neg(\varphi')) \cap D_k = \emptyset$. From above we have $V(neg(\varphi)) \cap M = \emptyset$ and thus $V(neg(\varphi)) \cap D_k^{\uparrow s} = \emptyset$ since $D_k^{\uparrow s} \subseteq M$. From this it follows that $V'(neg(\varphi')) \cap D_k = \emptyset$.

  Lastly, the nonequalities of $neq(\varphi') = neq(\varphi)$ must also be satisfied under V', because they are satisfied under V. Therefore, during global transition k, during the computation of the stratum of $\boldsymbol{f}$'s predicate in D_k, the rule φ' under valuation V' derives $\boldsymbol{f}^{\downarrow} \in D_k$ and thus $\boldsymbol{f} \in D_k^{\uparrow s} \subseteq M_k \subseteq M$.

- Suppose that ψ is inductive. There must be a fact $\texttt{tsucc}(s-1, s) \in pos(\psi)$ and thus $s > 0$. Because ψ is in $G_M(P)$, we can choose an original inductive rule φ from P and a valuation V for φ such that $V(head(\varphi)) = head(\psi)$, $V(pos(\varphi)) = pos(\psi)$, $V(neq(\varphi)) = neq(\psi)$ and $V(neg(\varphi)) \cap M = \emptyset$. Valuation V assigns the value $s-1$ to the body timestamp variable and s to the head timestamp-variable of φ. Let φ' denote the simplication of φ and let V' denote the accompanying simplification of V. Because $\mathcal{R}$ is fair, $\boldsymbol{f}$ is well-formedand $s-1 \in \mathbb{N}$, by Lemma B.3 we can consider the global transition k so that its recipient x_k is x and $local_{\mathcal{R}}(k) = s-1$. We show that V' is satisfiable for φ' during global transition k.

  We first show that $V'(pos(\varphi')) \subseteq D_k$. This is similar as in the deductive case above, but we write it down because it is also slightly different. By definition of φ', we have $pos(\varphi') = (pos(\varphi)|_{sch(P)})^{\downarrow}$. Since ψ produced $\boldsymbol{f}$, we have $V(pos(\varphi)) \subseteq N_{i-1}$ and thus $V(pos(\varphi)) \subseteq M$ by applying the induction hypothesis. Specifically, $V(pos(\varphi)|_{sch(P)}) \subseteq M$. The facts in $V(pos(\varphi)|_{sch(P)})$ are over the schema $sch(P)^{\mathrm{LT}}$, have location specifier x and timestamp $s-1$. By Lemma B.3, it follows that the set (5.1) is the only part of M that contains facts over schema $sch(P)^{\mathrm{LT}}$ with location specifier x and timestamp $s-1$. Therefore $V(pos(\varphi)|_{sch(P)}) \subseteq D_k^{\uparrow s-1}$. From this it follows that $V'(pos(\varphi')) \subseteq D_k$.

  Showing that $V'(neg(\varphi')) \cap D_k = \emptyset$ is similarly as in the deductive case above. Also, by definition of φ', $neq(\varphi') = neq(\varphi)$. Therefore, overall, the rule φ' under valuation V' derives $\boldsymbol{f}^{\downarrow} \in I_k$ (the inductively derived facts during global transition k). Thus $\boldsymbol{f}^{\downarrow} \in s^{\rho_{k+1}}|^{x}$ by the operational semantics. Let $j > k$ be the global transition index such that $x_j = x_k$ and $k \in prev_{\mathcal{R}}(j)$, so j is the first global transition after k in which x is the recipient again. We have

$$
\begin{aligned}
\boldsymbol{f} &\in I_k^{\uparrow s} \subseteq (s^{\rho_{k+1}}|^{x})^{\uparrow s} = (s^{\rho_j}|^{x})^{\uparrow s} \\
&\subseteq S_j^{\uparrow s} \subseteq D_j^{\uparrow s} \subseteq M_j \subseteq M.
\end{aligned}
$$

- Suppose that ψ is of the form (4.11):

$$
R(x, s, \bar{a}) \leftarrow R_{\mathrm{snd}}(y, t, x, s, \bar{a}),
$$

  for some values $y \in \mathcal{N}$ and $t \in \mathbb{N}$. We have $R_{\mathrm{snd}}(y, t, x, s, \bar{a}) \in N_{i-1}$ and thus $R_{\mathrm{snd}}(y, t, x, s, \bar{a}) \in M$ by applying the induction hypothesis. By construction of M, there must be some global transition j such that $R_{\mathrm{snd}}(y, t, x, s, \bar{a}) \in M_j \setminus M_{j-1}$. By construction of the set (5.6) for j, this implies recipient x_j is y, $local_{\mathcal{R}}(j) = t$ and $R(x, \bar{a}) \in \delta_j$. Furthermore, $s = local_{\mathcal{R}}(k)$ with $k = \alpha(j, R(x, \bar{a}))$. By definition of α, this then implies $R(x, \bar{a}) \in untag(m_k) \subseteq S_k \subseteq D_k$. Thus $R(x, s, \bar{a}) \in D_k^{\uparrow s} \subseteq M_k \subseteq M$ by definition of the set (5.1) for k.

Relation R_{snd}.

Suppose $\boldsymbol{f}$ is an R_{snd}-fact, which is of the form $R_{\mathrm{snd}}(x, s, y, t, \bar{a})$, where R is a relation name in $sch(P)$. Let ψ in $G_M(P)$ be a ground rule that produced $\boldsymbol{f}$.

We first show that there is a global transition k in $\mathcal{R}$ with $R(y, \bar{a}) \in \delta_k$. First, because ψ exists in $G_M(P)$, we can choose a non-ground rule φ in $pure(P)$ and a valuation V for φ on which ψ is based, with specifically $V(neg(\varphi)) \cap M = \emptyset$. Rule φ has head-predicate R_{snd} and is based on the form (4.8). Let φ_2 be the original asynchronous rule of P on which φ in turn is based, which is of the following form:

$$
\begin{aligned}
R(y', t', \bar{v}) \quad \leftarrow \quad & \mathbf{B}\{x', s' \mid \bar{v}, \bar{w}, y'\}, \texttt{time}(t'), \\
& choice(\langle x', s', y', \bar{v} \rangle, \langle t' \rangle);
\end{aligned}
$$

where $\mathbf{B}$ is a sequence of literals and nonequalities, where x' and s' are the variables for the location specifier and timestamp of the body respectively. The valuation V can be used as a valuation of φ_2 because, by definition of φ out of φ_2, the variables of φ_2 are included in those of φ. Let φ' be the simplified version of φ_2. Let V' be the restriction of V to $vars(\varphi')$. Because $\boldsymbol{f}$ is well-formed, by Lemma B.3, there is a unique global transition k of $\mathcal{R}$ such that the recipient x_k is x and $s = local_{\mathcal{R}}(k)$.

We show that V' is satisfying for φ' during global transition k. We first show that $V'(pos(\varphi')) \subseteq D_k$. Since ψ derives $\boldsymbol{f}$, we know that $pos(\psi) \subseteq N_{i-1}$ and thus $pos(\psi) \subseteq M$ by applying the induction hypothesis. Therefore $V(pos(\varphi)) \subseteq M$ since $V(pos(\varphi)) = pos(\psi)$. By definition of φ out of φ_2, we have $pos(\varphi_2) \subseteq pos(\varphi)$, and thus $V(pos(\varphi_2)) \subseteq M$. The facts in $V(pos(\varphi_2)|_{sch(P)})$ are over the schema $sch(P)^{\mathrm{LT}}$, have location specifier x and timestamp s. By Lemma B.3, the set (5.1) for k is the only part of M where we add facts over $sch(P)^{\mathrm{LT}}$ with location specifier x and timestamp s. Thus $V(pos(\varphi_2)|_{sch(P)}) \subseteq D_k^{\uparrow s}$. By definition of φ' we have $pos(\varphi') = (pos(\varphi_2)|_{sch(P)})^{\downarrow}$. From this it follows that $V'(pos(\varphi')) \subseteq D_k$.

Now we show that $V'(neg(\varphi')) \cap D_k = \emptyset$. From above we have $V(neg(\varphi)) \cap M = \emptyset$ and thus $V(neg(\varphi)) \cap D_k^{\uparrow s} = \emptyset$ because $D_k^{\uparrow s} \subseteq M$. Since $neg(\varphi_2) = neg(\varphi)$ and $neg(\varphi') \subseteq neg(\varphi_2)^{\downarrow}$, it follows that $V'(neg(\varphi')) \cap D_k = \emptyset$. The nonequalities of φ' must also be satisfied under V'. Alltogether, this causes φ' to produce the fact $R(y, \bar{a}) \in \delta_k$ under valuation V'.

Now we show that $R_{\mathrm{snd}}(x, s, y, t, \bar{a}) \in M$. The rule φ is of the form (4.8). There must be a fact $\texttt{chosen}_R(x, s, y, \bar{a}, t) \in pos(\psi)$, which is in M by the induction hypothesis. By Lemma (B.3), this $\texttt{chosen}_R$-fact is more specifically in the set (5.7) of global transition k. By definition of this set, we have $t = local_{\mathcal{R}}(j)$ where $j = \alpha(k, R(y, \bar{a}))$. Then, by definition of the set (5.6) for k, we have $R_{\mathrm{snd}}(x, s, y, t, \bar{a}) \in M_k \subseteq M$.

Relation `rcvClock`.

Suppose $\boldsymbol{f}$ is a `rcvClock`-fact. There are several ways in which it can have been generated.

- By a ground rule of the form (4.3):

  $$\texttt{rcvClock}(x, s, y, s) \leftarrow \texttt{all}(x), \texttt{all}(y), x \neq y, \texttt{zero}(s).$$

  By the induction hypothesis, M contains $\texttt{all}(x)$, $\texttt{all}(y)$ and $\texttt{zero}(s)$. Because M is well-formed by construction, this implies $x, y \in \mathcal{N}$ and $s \in \mathbb{N}$. By construction of M, $\texttt{zero}(s) \in M_{-1}$ and thus $s = 0$. Now, by construction of the set M_{-1} we now have $\texttt{rcvClock}(x, 0, y, 0) \in M_{-1} \subseteq M$.

- By a ground rule of the form (4.4):

  $$\texttt{rcvClock}(x, s, x, s') \leftarrow \texttt{all}(x), \texttt{tsucc}(s, s').$$

  By the induction hypothesis, M and more specifically $decl(H)$ contains $\texttt{all}(x)$ and $\texttt{tsucc}(s, s')$. This implies we have $s, s' \in \mathbb{N}$ and $s' = s + 1$. By Lemma B.3 there is a unique global transition k such that the recipient x_k is x and $s = local_{\mathcal{R}}(k)$. Now by definition of the set (5.2) for k, we have $\texttt{rcvClock}(x, s, x, s') \in M_k \subseteq M$.

- By a ground rule of the form (4.5):

  $$\begin{aligned} \texttt{rcvClock}(x, s', y, t) \quad \leftarrow \quad & \texttt{clock}(x, s, y, t), x \neq y, \\ & \texttt{tsucc}(s, s'). \end{aligned}$$

  By the induction hypothesis, M contains $\texttt{clock}(x, s, y, t)$ and $\texttt{tsucc}(s, s')$. Because M is well-formed by construction, this implies $x, y \in \mathcal{N}$ and $s, t \in \mathbb{N}$. Since $\texttt{tsucc}(s, s') \in decl(H)$ we have $s' \in \mathbb{N}$ and $s' = s + 1$. Additionally, the ground rule above implies $x \neq y$. By Lemma B.3 there is a unique global transition k such that the recipient x_k is x and $s' = local_{\mathcal{R}}(k)$. Similarly there is a unique global transition j where the recipient x_j is also x and $s = local_{\mathcal{R}}(j)$. Because $local_{\mathcal{R}}(j) + 1 = local_{\mathcal{R}}(k)$ we have $j < k$ and more specifically $j \in prev_{\mathcal{R}}(k)$. Because the set (5.5) for global transition j is the only part of M where we add $\texttt{clock}$-facts with first components x and s, we have $\texttt{clock}(x, s, y, t) \in M_j$. By definition of the set (5.5) for j, we have $v_{\mathcal{R}}(j)[y] = t$. Now, since $x \neq y$, we have $t \in \pi_{\mathcal{R}}(k, y)$. And by definition of the set (5.3) for global transition k, we obtain $\texttt{rcvClock}(x, s', y, t) \in M_k$.

- By a rule of the form (4.12):

  $$\texttt{rcvClock}(y, t, z, u) \quad \leftarrow \quad R_{\mathrm{snd}}(x, s, y, t, \bar{a}), \texttt{clock}(x, s, z, u).$$

  Then by applying the induction hypothesis both body facts are in M. By construction of M, there must be a global transition k such that $R_{\mathrm{snd}}(x, s, y, t, \bar{a}) \in M_k \setminus M_{k-1}$. More specifically, $R_{\mathrm{snd}}(x, s, y, t, \bar{a})$ is in the set (5.6) for k. This implies $x = x_k$, $s = local_{\mathcal{R}}(k)$, $R(y, \bar{a}) \in \delta_k$ and $t = local_{\mathcal{R}}(j)$ with $j = \alpha(k, R(y, \bar{a}))$. Furthermore, $\texttt{clock}(x, s, z, u)$ is in the set (5.5) for k because by Lemma (B.3) this is the only part of M where we add $\texttt{clock}$-facts with the first two components x and s. By definition of the set (5.5) for k , we have $z \in \mathcal{N}$ and $v_{\mathcal{R}}(k)[z] = u$. Now by definition of the set (5.9) for k we have $\texttt{rcvClock}(y, t, z, u) \in M$.

Relation `isBehind`.

Suppose $\boldsymbol{f}$ is an `isBehind`-fact. The ground rule that has derived $\boldsymbol{f}$ is of the form (4.6):

$$\begin{aligned} \texttt{isBehind}(x, s, y, t) \quad \leftarrow \quad & \texttt{rcvClock}(x, s, y, t), \\ & \texttt{rcvClock}(x, s, y, t'), t' < t. \end{aligned}$$

By applying the induction hypothesis, we have $\texttt{rcvClock}(x, s, y, t) \in M$ and $\texttt{rcvClock}(x, s, y, t') \in M$. Because M is well-formed by construction, we have $x, y \in \mathcal{N}$ and $s, t, t' \in \mathbb{N}$. by Lemma (B.3) there is a global transition k of $\mathcal{R}$ such that the recipient x_k is x and $s = local_{\mathcal{R}}(k)$. By Lemma E.8 we have $\texttt{rcvClock}(x, s, y, t) \in M_k$ and $\texttt{rcvClock}(x, s, y, t') \in M_k$.

Now since $\texttt{rcvClock}(x,s,y,t) \in M_k$, $\texttt{rcvClock}(x,s,y,t') \in M_k$ and $t' < t$, by definition of the set (5.4) for k, we have $\texttt{isBehind}(x,s,y,t) \in M_k$.

Relation clock.

Suppose $\boldsymbol{f}$ is a clock-fact. Let ψ be the ground rule, based on the form (4.7), that derived $\boldsymbol{f}$:

$$\texttt{clock}(x,s,y,t) \leftarrow \texttt{rcvClock}(x,s,y,t).$$

By applying the induction hypothesis, we must have $\texttt{rcvClock}(x,s,y,t) \in N_{i-1} \subseteq M$. Because M is well-formed by construction, by Lemma B.3 there is a unique global transition k of $\mathcal{R}$ such that $x_k = x$ and $s = \mathit{local}_{\mathcal{R}}(k)$. Since $\texttt{rcvClock}(x,s,y,t) \in M$, by Lemma E.8 we have $\texttt{rcvClock}(x,s,y,t) \in M_k$.

As mentioned above, the non-ground rule φ in $\mathit{pure}(P)$ on which ψ is based is of the form (4.7):

$$\texttt{clock}(x's',y',t') \leftarrow \texttt{rcvClock}(x',s',y,'t'), \neg\texttt{isBehind}(x',s',y',t').$$

Since $\psi \in G_M(P)$ we must have $\texttt{isBehind}(x,s,y,t) \notin M$ and thus $\texttt{isBehind}(x,s,y,t) \notin M_k$ since $M_k \subseteq M$. By definition of the set (5.4) for k, this implies that for all facts $\texttt{rcvClock}(x,s,y,v) \in M_k$ we have $v \leq t$.

Denote $u = v_{\mathcal{R}}(k)[y]$. Now we show that $t = u$. First, we show that $\texttt{rcvClock}(x,s,y,u) \in M_k$. If $u = 0$ then surely $u \leq t$ because $t \in \mathbb{N}$. Now suppose $u > 0$. We show that $\texttt{rcvClock}(x,s,y,u) \in M_k$. If $u > 0$ it is not possible that $\texttt{rcvClock}(x,s,y,u) \in M_{-1}$.

- Suppose $y = x$. By definition $u = \mathit{local}_{\mathcal{R}}(k) + 1$. By definition of the set (5.2) for k we have $\texttt{rcvClock}(x,s,y,u) \in M_k$.
- Suppose $y \neq x$. By definition $u = \max(\{0\} \cup \mu_{\mathcal{R}}(k,y) \cup \pi_{\mathcal{R}}(k,y))$. Suppose $u \in \mu_{\mathcal{R}}(k,y)$. Then by Lemma E.7 we have $\texttt{rcvClock}(x,s,y,u) \in M_k$. Suppose $u \in \pi_{\mathcal{R}}(k,y)$. Then by definition of the set (5.3) for M_k we have $\texttt{rcvClock}(x,s,y,u) \in M_k$.

For each fact $\texttt{rcvClock}(x,s,y,v) \in M_k$ we have $v \leq t$ by our reasoning above. Therefore $u \leq t$.

Now we show that $u = t$. We already have $u \leq t$. We show that $t \leq u$. Abbreviate $\boldsymbol{g} = \texttt{rcvClock}(x,s,y,t)$. We look at the options where $\boldsymbol{g}$ can come from:

- Suppose $\boldsymbol{g}$ is in the set (5.2) for k. Then $x = y$ and $t = s + 1 = u$ by definition of u.
- Suppose that $\boldsymbol{g}$ is in the set (5.3) for k. Then $t \in \pi_{\mathcal{R}}(k,y)$ and then $t \leq u$ by definition of u.
- Fact $\boldsymbol{g}$ can not be in the set (5.9) for k because the second component should then be at least $s+1$ by Lemma (D.4).
- Suppose that $\boldsymbol{g} \in M_{-1}$. Then $t = 0$ and $t \leq u$ immediately holds.
- Suppose that there is a global transition $j < k$ such that $\texttt{rcvClock}(x,s,y,t) \in M_j \setminus M_{j-1}$. The sets (5.2) and (5.3) for j cannot contain $\texttt{rcvClock}(x,s,y,t)$: if $x \neq x_j$ then the first component would have to be different from x, and if $x = x_j$ then $\mathit{local}_{\mathcal{R}}(j) < \mathit{local}_{\mathcal{R}}(k) = s$. Therefore, only the set (5.9) for j can contain $\texttt{rcvClock}(x,s,y,t)$. By construction of this set, we have $v_{\mathcal{R}}(j)[y] = t$, and there is a fact $\boldsymbol{g} \in \delta_j$ and a global transition $h = \alpha(j,\boldsymbol{g})$ such that $j < h$, $x_h = x$, $\mathit{local}_{\mathcal{R}}(h) = s$. Then $h = k$ by Lemma B.3. If $y = x$ then $v_{\mathcal{R}}(j)[y] < v_{\mathcal{R}}(k)[y]$ by Lemma D.3 and thus $t < u$. If $y \neq x$ then $v_{\mathcal{R}}(j)[y] \in \mu_{\mathcal{R}}(k,y)$ by definition of $\mu_{\mathcal{R}}(k,y)$, and thus $t \leq u$ by definition of u as $v_{\mathcal{R}}(k)[y]$.

From $u \leq t$ and $t \leq u$ we obtain that $u = t$. Now by construction of the set (5.5) for k we have $\texttt{clock}(x,s,y,t) \in M_k \subseteq M$.

Relation $\texttt{chosen}_R$.

Suppose $\boldsymbol{f}$ is a $\texttt{chosen}_R$-fact with R a relation in $\mathit{sch}(P)$. Then $\boldsymbol{f}$ was generated by a ground rule ψ, based on the form (4.9):

$$\begin{aligned}\texttt{chosen}_R(x,s,y,\bar{a},t) \quad \leftarrow \quad & \mathbf{B}\{x,s \mid \bar{a},\bar{b},y\}, \texttt{all}(y), \\ & \texttt{clock}(x,s,y,u), \texttt{time}(t), u \leq t;\end{aligned}$$

where the (ground) literals in $\mathbf{B}\{x,s \mid \bar{a},\bar{b},y\}$ are over $\mathit{sch}(P)$. The values x and s are the location specifier and timestamp of the body facts over schema $\mathit{sch}(P)^{\mathrm{LT}}$ respectively. This location specifier and timestamp are used for all body facts over schema $\mathit{sch}(P)^{\mathrm{LT}}$. There must be a non-ground rule φ in $\mathit{pure}(P)$ and a valuation V on which ψ is based, of the form 4.9. In turn, there must be an original asynchronous rule φ_2 in P such that φ is obtained from φ by applying the form (4.9). The head predicate of φ_2 is R. Because the positive variables of φ_2 are included in those of φ by construction of φ, we can apply V to φ_2 as well. Let φ' be the simplification of φ_2 and let V' be the restriction of V to the variables of φ'.

Because M is well-formed, by Lemma B.3 there is a unique global transition k of $\mathcal{R}$ so that the recipient x_k is x and $s = \mathit{local}_{\mathcal{R}}(k)$. Similarly to the case where $\boldsymbol{f}$ is a fact with predicate R_{snd} (see earlier), we can show that V' is satisfying for φ' during global transition k and we obtain that $R(y,\bar{a}) \in \delta_k$.

Since ψ exists in $G_M(P)$, we have $V(\mathit{neg}(\varphi)) \cap M = \emptyset$ and specifically using the form (4.9), we have $\texttt{other}_R(x,s,y,\bar{a},t) \notin M$. Since the set (5.8) for k is the only part of M that can contain $\texttt{other}_R$-facts with first two components x and s, we have that $\texttt{other}_R(x,s,y,\bar{a},t)$ is not in this set, which could be for the following reasons (looking at the definition of the set (5.8)):

- Suppose $R(y,\bar{a}) \notin \delta_k$. That is not possible (see earlier);
- Suppose $t \notin \mathbb{N}$. That is is not possible because $\texttt{time}(t) \in \mathit{decl}(H)$ and $\mathit{decl}(H)$ is well-formed;

- Suppose $t < v_{\mathcal{R}}(k)[y]$. Referring to ground rule ψ above, we have $\texttt{clock}(x, s, y, u) \in N_{i-1} \subseteq M$ by the induction hypothesis. By Lemma B.3, the set (5.5) for k is the only part of M where we add $\texttt{clock}$-facts with first components x and s. Thus $\texttt{clock}(x, s, y, u)$ is in the set (5.5) for k and thus $v_{\mathcal{R}}(k)[y] = u$. Also, since according to ground rule ψ above we have $u \leq t$, we cannot have $t < v_{\mathcal{R}}(k)[y]$.
- Suppose $t = \mathit{local}_{\mathcal{R}}(j)$ with $j = \alpha(k, R(y, \bar{a}))$. This is the only case that remains, so it must be true.

Now by using the definition of the set (5.7) for k, we have $\texttt{chosen}_R(x, s, y, \bar{a}, t) \in M_k \subseteq M$.

Relation $\texttt{other}_R$.

Suppose $\boldsymbol{f}$ has predicate $\texttt{other}_R$ with R a relation in $\mathit{sch}(P)$. The fact was generated by a ground rule ψ, based on the form (4.10):

$$\begin{aligned} \texttt{other}_R(x, s, y, \bar{a}, t) \quad \leftarrow \quad & \mathbf{B}\{x, s \mid \bar{a}, \bar{b}, y\}, \texttt{all}(y), \\ & \texttt{clock}(x, s, y, u), \texttt{time}(t), u \leq t, \\ & \texttt{chosen}_R(x, s, y, \bar{a}, t'), t \neq t'. \end{aligned}$$

We use a similar notation as for the ground rule in the reasoning about the $\texttt{chosen}_R$-fact above. Because M is well-formed, by Lemma B.3 there is a unique global transition k such that the recipient x_k is x and $s = \mathit{local}_{\mathcal{R}}(k)$. We can show similarly as for the $\texttt{chosen}_R$-fact above, that $R(y, \bar{a})$ is sent during global transition k, formally, $R(y, \bar{a}) \in \delta_k$.

In addition, by applying the induction hypothesis, $\texttt{chosen}_R(x, s, y, \bar{a}, t') \in N_{i-1} \subseteq M$. Because the set (5.7) for global transition k is the only part of M where we add $\texttt{chosen}_R$-facts with first components x and s, we have that $\texttt{chosen}_R(x, s, y, \bar{a}, t')$ is in this set, and thus $t' = \mathit{local}_{\mathcal{R}}(j)$ with $j = \alpha(k, R(y, \bar{a}))$. Similarly, the fact $\texttt{clock}(x, s, y, u) \in N_{i-1} \subseteq M$ implies that $\texttt{clock}(x, s, y, u)$ is in the set (5.5) for k and thus $v_{\mathcal{R}}(k)[y] = u$. The rule ψ above then implies $t \in \mathbb{N}$, $v_{\mathcal{R}}(k)[y] \leq t$ and $t \neq t'$. Now, by definition of the set 5.8 for global transition k, we have $\texttt{other}_R(x, s, y, \bar{a}, t) \in M_k \subseteq M$.

Relation $\texttt{notZero}$.

Suppose that $\boldsymbol{f}$ is of the form $\texttt{notZero}(s)$. Then $\boldsymbol{f}$ is derived by a ground rule, based on (4.1):

$$\texttt{notZero}(s) \leftarrow \texttt{tsucc}(t, s).$$

Since $\texttt{tsucc}(t, s) \in \mathit{decl}(H)$ implies $s > 0$, we have $\texttt{notZero}(s) \in M_{-1} \subseteq M$.

Relation $\texttt{zero}$.

Suppose that $\boldsymbol{f}$ is of the form $\texttt{zero}(s)$. Then $\boldsymbol{f}$ is derived by a ground rule of the following form, based on (4.2):

$$\texttt{zero}(s) \leftarrow \texttt{time}(s).$$

Because this ground rule exists in $G_M(P)$, we have $\texttt{notZero}(s) \notin M$ and thus $\texttt{notZero}(s) \notin M_{-1}$. But $\texttt{time}(s) \in \mathit{decl}(H)$ implies $s \in \mathbb{N}$ and by definition of the set M_{-1} this implies $s = 0$. Furthermore, we have $\texttt{zero}(s) \in M_{-1}$, again by definition of M_{-1}. □

Lemma E.7. *Consider Setting E.1. Let i be a global transition of $\mathcal{R}$. Denote $x = x_i$ and $s = \mathit{local}_{\mathcal{R}}(i)$. Let $z \in \mathcal{N}$ with $z \neq x$. Let $u \in \mu_{\mathcal{R}}(i, z)$. We have $\texttt{rcvClock}(x, s, z, u) \in M_i$.*

Proof. By definition of $\mu_{\mathcal{R}}(i, z)$ there is a global transition $k < i$ and a fact $\boldsymbol{f} \in \delta_k$ with $\alpha(k, \boldsymbol{f}) = i$ and $v_{\mathcal{R}}(k)[z] = u$. The fact $\boldsymbol{f}$ is of the form $R(x, \bar{a})$. Abbreviate $l = \mathit{local}_{\mathcal{R}}(k)$. By the set (5.9) for global transition k, we have $\texttt{rcvClock}(x, s, z, u) \in M_k$. □

Lemma E.8. *Consider Setting E.1. Consider a fact $\texttt{rcvClock}(x, s, y, t) \in M$. Because M is well-formed, we have $x \in \mathcal{N}$ and $s \in \mathbb{N}$. Then by Lemma B.3 there is a unique global transition i of $\mathcal{R}$ such that $x_i = x$ and $\mathit{local}_{\mathcal{R}}(i) = s$. We have $\texttt{rcvClock}(x, s, y, t) \in M_i$.*

Proof. Abbreviate $\boldsymbol{f} = \texttt{rcvClock}(x, s, y, t)$. First suppose $\boldsymbol{f} \in M_{-1}$. Then we immediately have $\boldsymbol{f} \in M_i$ because $M_{-1} \subseteq M_i$ by definition of M_i.

Now suppose $\boldsymbol{f} \notin M_{-1}$. As a proof by contradiction, suppose now that $\boldsymbol{f} \notin M_i$. Because $\boldsymbol{f} \in M$, there must be a global transition j with $\boldsymbol{f} \in M_j \setminus M_{j-1}$. We must have $i < j$ because otherwise $M_j \subseteq M_i$ and $\boldsymbol{f} \in M_i$. The sets (5.2) and (5.3) for global transition j cannot contain $\boldsymbol{f}$:

- If $x_j \neq x$ then the first component of $\boldsymbol{f}$ should be different from x.
- If $x_j = x$ then the second component of $\boldsymbol{f}$ should be $\mathit{local}_{\mathcal{R}}(j)$, which is different from s since $\mathit{local}_{\mathcal{R}}(i) < \mathit{local}_{\mathcal{R}}(j)$, which follows from $i < j$ and the definition of $\mathit{local}_{\mathcal{R}}$.

The fact $\boldsymbol{f}$ can only be in the set (5.9) for j. Now, by definition of the set (5.9) for j, there must be a fact $\boldsymbol{g} \in \delta_j$ of the form $R(x, \bar{a})$ and a global transition h such that $h = \alpha(j, \boldsymbol{g})$ and $s = \mathit{local}_{\mathcal{R}}(h)$. But by definition of α we have $j < h$ and $x_h = x$. By Lemma (B.3) we have $i = h$. But then $j < i$ is a contradiction.

Therefore $\texttt{rcvClock}(x, s, y, t) \in M_i$. □

Proposition E.9. *Consider Setting E.1. The trace M of $\mathcal{R}$ is a fair stable model of pure(P) on input decl(H).*

Proof. First, using Lemmas E.2 and E.6 we know that $M = N$ and thus M is stable.

Now we show that M is fair as well. Let $(x, s) \in \mathcal{N} \times \mathbb{N}$. Fix a relation R in $sch(P)$ for which the relation R_{snd} occurs in $pure(P)$. We must show that the number of R_{snd}-facts in M with x and s as third and fourth components is finite. First, by Lemma B.3 there is one unique global transition k of $\mathcal{R}$ such that the recipient x_k is x and $local_{\mathcal{R}}(k) = s$.

Consider a fact $R_{\text{snd}}(y, t, x, s, \bar{a}) \in M$. We first show that $R_{\text{snd}}(y, t, x, s, \bar{a}) \in M_k$. By construction of M there must be a global transition j such that $R_{\text{snd}}(y, t, x, s, \bar{a}) \in M_j \setminus M_{j-1}$. By definition of the set (5.6) for j, we have $x_j = y$, $t = local_{\mathcal{R}}(j)$, there is a fact $R(x, \bar{a}) \in \delta_j$ and a global transition h with $j < h$ such that $h = \alpha(j, R(x, \bar{a}))$, $x_h = x$ and $s = local_{\mathcal{R}}(h)$. By Lemma B.3 this implies $h = k$. Therefore $j < k$. And since $M_j \subseteq M_k$, we have $R_{\text{snd}}(y, t, x, s, \bar{a}) \in M_k$.

As an intermediate conclusion, all R_{snd}-facts in M with x and s as third and fourth components are generated by global transitions smaller than k. Let j be such a global transition, like above. By Lemma B.4, the set δ_j is finite. Therefore, the number of R_{snd}-facts contributed by the set (5.6) for j is finite. Now, because there are only a finite number of global transitions that came before global transition k, there are only a finite number of R_{snd}-facts in M_k with x and s as third and fourth components. We conclude that M is fair. □

F. STABLE MODEL TO RUN

In this section we show that for every fair stable model of a Dedalus program on a distributed input database there exists a fair run whose trace is that stable model.

Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Let M be a *fair* stable model of $pure(P)$ on input $decl(H)$. All lemmas mentioned in this section are relative to this setting.

As a general property, because $decl(H)$ is well-formed by construction and because M is the output of $G_M(P)$ on input $decl(H)$, by Lemma C.1 we have that M is well-formed.

We reuse the notations from the previous sections.

F.1 Time properties

The following properties give insight in what clock information is represented by M.

Lemma F.1. *For $t \in \mathbb{N} \setminus \{0\}$ we have $\texttt{notZero}(t) \in M$ and $\texttt{zero}(t) \notin M$. Also, $\texttt{zero}(0) \in M$.*

Proof. Let $t \in \mathbb{N} \setminus \{0\}$. By definition of $decl(H)$ we have $\texttt{tsucc}(t-1, t) \in decl(H) \subseteq M$. Therefore, the fact $\texttt{notZero}(t)$ is derived by the following ground rule in $G_M(P)$, based on the rule (4.1):

$$\texttt{notZero}(t) \leftarrow \texttt{tsucc}(t-1, t).$$

The only rule in $pure(P)$ with head-predicate $\texttt{zero}$ is (4.2). Because $\texttt{notZero}(t) \in M$ the following ground rule cannot exist in $G_M(P)$:

$$\texttt{zero}(t) \leftarrow \texttt{time}(t).$$

Therefore $\texttt{zero}(t) \notin M$.

The fact $\texttt{notZero}(0)$ cannot be derived with ground rules of the form (4.1) because there is no fact $\texttt{tsucc}(s, 0) \in decl(H)$, therefore $\texttt{notZero}(0) \notin M$. Thus, the following ground rule, based on the rule (4.2), exists in $G_M(P)$ and it derives $\texttt{zero}(0) \in M$:

$$\texttt{zero}(0) \leftarrow \texttt{time}(0).$$

□

Lemma F.2. *For $x, y \in \mathcal{N}$ with $x \neq y$ we have $\texttt{rcvClock}(x, 0, y, 0) \in M$.*
For $x \in \mathcal{N}$ and $s \in \mathbb{N}$ we have $\texttt{rcvClock}(x, s, x, s+1) \in M$.

Proof. First, by definition, $decl(H)$ contains for each $x \in \mathcal{N}$ the fact $\texttt{all}(x)$ and for each $s \in \mathbb{N}$ the fact $\texttt{tsucc}(s, s+1)$. Also, from Lemma F.1 we know that there is a fact $\texttt{zero}(0) \in M$. Now, the facts of this lemma are derived by ground rules of the forms (4.3) and (4.4), because M is a fixpoint:

$$\texttt{rcvClock}(x, s, y, s) \leftarrow \texttt{all}(x), \texttt{all}(y), x \neq y, \texttt{zero}(s).$$

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

□

Lemma F.3. *For $x, y \in \mathcal{N}$ and $s \in \mathbb{N}$, there is at most one value t such that $\texttt{clock}(x, s, y, t) \in M$.*

Proof. Suppose that there are two different values t and t' such that $\texttt{clock}(x, s, y, t) \in M$ and $\texttt{clock}(x, s, y, t') \in M$. Because M is well-formed, we have $t, t' \in \mathbb{N}$. The only rule in *pure*(P) for deriving clock-facts is the rule (4.7). This implies that $\texttt{rcvClock}(x, s, y, t) \in M$ and $\texttt{rcvClock}(x, s, y, t') \in M$. Assume without loss of generality that $t < t'$. Then $\texttt{isBehind}(x, s, y, t) \in M$ is derived by a ground rule based on the form (4.6) because M is a fixpoint. But then the following ground rule cannot exist in $G_M(P)$, based on the rule (4.7):

$$\texttt{clock}(x, s, y, t) \leftarrow \texttt{rcvClock}(x, s, y, t).$$

Therefore $\texttt{clock}(x, s, y, t) \notin M$, a contradiction. □

Lemma F.4. *Let* $(x, s) \in \mathcal{N} \times \mathbb{N}$. *The following subset of* M *is finite:*

$$\{\texttt{clock}(z, u, y, t) \in M \mid (z, u) = (x, s)\}.$$

Proof. First, because M is well-formed there can be no fact $\texttt{clock}(x, s, y, t) \in M$ with $y \notin \mathcal{N}$. Also, for a node $y \in \mathcal{N}$, Lemma F.3 tells us that there is at most one value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$. Now since $\mathcal{N}$ is finite, overall there are only a finite number of clock-facts in M with x and s as first and second components. □

Lemma F.5. *Let* $(x, s) \in \mathcal{N} \times \mathbb{N}$. *The following subset of* M *is finite:*

$$\{\texttt{rcvClock}(z, u, y, t) \in M \mid (z, u) = (x, s)\}.$$

Proof. First, suppose that rcvClock-facts in M could only be generated by ground rules in $G_M(P)$ that are based on the forms (4.3), (4.4) and (4.5):

- Ground rules of the form (4.3) cannot cause the property of this lemma to be violated because the second and fourth component of the generated rcvClock-facts must be the same and because $\mathcal{N}$ is finite, which implies that there are only a finite number of all-facts in *decl*(H).
- Ground rules of the form (4.4) cannot cause the property to be violated because if $\texttt{tsucc}(s, s') \in decl(H)$ and $\texttt{tsucc}(s, s'') \in decl(H)$ then $s' = s''$ (functional dependency).
- Ground rules of the form (4.5) cannot cause the property to be violated because *(i)* if $\texttt{tsucc}(s', s) \in decl(H)$ and $\texttt{tsucc}(s'', s) \in decl(H)$ then $s' = s''$ (functional dependency) and *(ii)* by Lemma F.4, for each $(y, t) \in \mathcal{N} \times \mathbb{N}$ there are only a finite number of clock-facts in M having y and t as first two components.

Now the only way the property could be violated is by ground rules of the form (4.12), where R is a relation in *sch*(P):

$$\texttt{rcvClock}(x', s', z', u') \quad \leftarrow \quad R_{\text{snd}}(y', t', x', s', \bar{v}), \texttt{clock}(y', t', z', u').$$

Let us first focus on a fact $R_{\text{snd}}(y, t, x, s, \bar{a}) \in M$. Because M is well-formed, we have $y \in \mathcal{N}$ and $t \in \mathbb{N}$. Let $z \in \mathcal{N}$. By Lemma F.4, there are only a finite number of clock-facts in M having y and t as first two components. Therefore, if we fix one R_{snd}-fact and combine it with clock-facts as indicated by the above rule, then only a finite number of rcvClock-facts can result with first components x and s. Now since M is fair, there are only a finite number of R_{snd}-facts in M with third and fourth components x and s, and thus ground rules of the form (4.12) cannot cause the property to be violated.

There are only a finite number of asynchronous rules in P and therefore there are only a finite number of rules of the form (4.12) in *pure*(P), and for each of these latter rules we have shown just above that they cannot cause the property to be violated. □

Lemma F.6. *Let* $x, y \in \mathcal{N}$ *and* $s \in \mathbb{N}$. *Suppose there is a value* $u \in \mathbb{N}$ *such that* $\texttt{rcvClock}(x, s, y, u) \in M$. *Then there is a value* $t \in \mathbb{N}$ *such that* $\texttt{clock}(x, s, y, t) \in M$.

Proof. Suppose that $\texttt{clock}(x, s, y, u) \notin M$. Because the only rule in *pure*(P) to derive clock-facts is (4.7), the following ground rule can thus not be available in $G_M(P)$:

$$\texttt{clock}(x, s, y, u) \leftarrow \texttt{rcvClock}(x, s, y, u).$$

This implies that $\texttt{isBehind}(x, s, y, u) \in M$. By rule (4.6), the existence of this isBehind-fact implies that there is some value u' such that $\texttt{rcvClock}(x, s, y, u') \in M$ and $u < u'$. Now similarly, $\texttt{clock}(x, s, y, u') \notin M$ would imply that there is yet another fact $\texttt{rcvClock}(x, s, y, u'') \in M$ such that $u' < u''$. But by Lemma F.5, this reasoning can't go on forever. Eventually we find the existence of a value t such that $\texttt{rcvClock}(x, s, y, t) \in M$ and $\texttt{isBehind}(x, s, y, t) \notin M$. Then the following ground rule, based on the rule (4.7), exists and derives $\texttt{clock}(x, s, y, t)$:

$$\texttt{clock}(x, s, y, t) \leftarrow \texttt{rcvClock}(x, s, y, t).$$

□

Lemma F.7. *Let* $x, y \in \mathcal{N}$ *and* $s \in \mathbb{N}$. *There exists at least one value* $t \in \mathbb{N}$ *such that* $\texttt{clock}(x, s, y, t) \in M$.

Proof. We show by induction on $s \in \mathbb{N}$ that there exists a value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$.

1. Base base ($s = 0$). If $x = y$ then $\texttt{rcvClock}(x, s, x, 1) \in M$ by Lemma F.2. If $x \neq y$ then $\texttt{rcvClock}(x, s, y, 0) \in M$, again by Lemma F.2. Thus there is some value u such that $\texttt{rcvClock}(x, s, y, u) \in M$. Then we can apply Lemma (F.6) to know that there is a value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$.
2. Induction hypothesis: assume that for $s \in \mathbb{N}$ there is a value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$.
3. Inductive step: we show that there is a value t such that $\texttt{clock}(x, s+1, y, t) \in M$. Like in the base case, if $x = y$ then $\texttt{rcvClock}(x, s+1, x, s+2) \in M$ by Lemma F.2. If $x \neq y$, we apply the induction hypothesis to know that there is some value u such that $\texttt{clock}(x, s, y, u) \in M$. Now, the following ground rule in $G_M(P)$, based on rule (4.5), derives $\texttt{rcvClock}(x, s+1, y, u) \in M$:

$$\begin{aligned} \texttt{rcvClock}(x, s+1, y, u) \quad \leftarrow \quad & \texttt{clock}(x, s, y, u), x \neq y, \\ & \texttt{tsucc}(s, s+1). \end{aligned}$$

   Thus, there is at least one value $v \in \mathbb{N}$ such that $\texttt{rcvClock}(x, s+1, y, v) \in M$. Now we can again apply Lemma (F.6) to obtain that there is a value $t \in \mathbb{N}$ such that $\texttt{clock}(x, s+1, y, t) \in M$.

□

Corollary F.8. *For $x, y \in \mathcal{N}$ and $s \in \mathbb{N}$ there is precisely one $t \in \mathbb{N}$ such that $\texttt{clock}(x, s, y, t) \in M$.*

Proof. This follows from Lemmas F.3 and F.7. □

F.2 Message sending

The following properties give insight in the message sending represented by M.

Lemma F.9. *For $R_{\text{snd}}(x, s, y, t, \bar{a}) \in M$ and $R_{\text{snd}}(x, s, y, t', \bar{a}) \in M$ we have $t = t'$ (functional dependency), with R a relation name in $sch(P)$.*

Proof. Proof by contradiction: suppose that $t \neq t'$. The facts $R_{\text{snd}}(x, s, y, t, \bar{a}) \in M$ and $R_{\text{snd}}(x, s, y, t', \bar{a}) \in M$ imply the existence of two ground rules to derive them, based on the form (4.8):

$$\begin{aligned} \psi_1 : \ R_{\text{snd}}(x, s, y, t, \bar{a}) \quad \leftarrow \quad & \mathbf{B}, \texttt{all}(y), \\ & \texttt{clock}(x, s, y, u), \texttt{time}(t), u \leq t, \\ & \texttt{chosen}_R(x, s, y, \bar{a}, t). \end{aligned}$$

$$\begin{aligned} \psi_2 : \ R_{\text{snd}}(x, s, y, t', \bar{a}) \quad \leftarrow \quad & \mathbf{B}', \texttt{all}(y), \\ & \texttt{clock}(x, s, y, u'), \texttt{time}(t'), u' \leq t', \\ & \texttt{chosen}_R(x, s, y, \bar{a}, t'). \end{aligned}$$

Here $\mathbf{B}$ and $\mathbf{B}'$ are lists of positive ground atoms (facts) and ground nonequalities. Since the above ground rules derived $R_{\text{snd}}(x, s, y, t, \bar{a}) \in M$ and $R_{\text{snd}}(x, s, y, t', \bar{a}) \in M$, the positive body facts in the above ground rules must be in M. Specifically $\texttt{chosen}_R(x, s, y, \bar{a}, t) \in M$ and $\texttt{chosen}_R(x, s, y, \bar{a}, t') \in M$. Also, by Lemma F.3 we have $u = u'$.

The presence of these $\texttt{chosen}_R$-facts in turn implies the presence of ground rules in $G_M(P)$ to derive them, based on the form (4.9):

$$\begin{aligned} \psi_3 : \ \texttt{chosen}_R(x, s, y, \bar{a}, t) \quad \leftarrow \quad & \mathbf{C}, \texttt{all}(y), \\ & \texttt{clock}(x, s, y, u), \texttt{time}(t), u \leq t. \end{aligned}$$

$$\begin{aligned} \psi_4 : \ \texttt{chosen}_R(x, s, y, \bar{a}, t') \quad \leftarrow \quad & \mathbf{C}', \texttt{all}(y), \\ & \texttt{clock}(x, s, y, u), \texttt{time}(t'), u \leq t'. \end{aligned}$$

Similarly to ψ_1 and ψ_2, here $\mathbf{C}$ and $\mathbf{C}'$ are lists of positive ground atoms and ground nonequalities. However, the original rules in *pure*(P) on which these ground rules are based contain negative $\texttt{other}_R$-literals in their body; see rule (4.9). Since ψ_2 and ψ_3 are in $G_M(P)$, the following facts must be absent from M: $\texttt{other}_R(x, s, y, \bar{a}, t)$ and $\texttt{other}_R(x, s, y, \bar{a}, t')$ (respectively).

The ground rules ψ_3 and ψ_4 are based on original rules φ and φ' respectively, that are in *pure*(P) and that are of the form (4.9). The rules φ and φ' in turn are based on original asynchronous rules φ_2 and φ'_2 respectively, that are in P. Applying the form (4.10) to φ_2 and φ'_2 respectively gives the following rules are in *pure*(P), where variable names are written in boldface and, for technical convenience, shared between the rules:

$$\begin{aligned} \varphi_3 : \ \texttt{other}_R(\mathbf{x}, \mathbf{s}, \mathbf{y}, \bar{\mathbf{v}}, \mathbf{t}) \quad \leftarrow \quad & pos(\varphi_2)|_{sch(P)}, neg(\varphi_2), neq(\varphi_2), \texttt{all}(\mathbf{y}), \\ & \texttt{clock}(\mathbf{x}, \mathbf{s}, \mathbf{y}, \mathbf{u}), \texttt{time}(\mathbf{t}), \mathbf{u} \leq \mathbf{t}, \\ & \texttt{chosen}_R(\mathbf{x}, \mathbf{s}, \mathbf{y}, \bar{\mathbf{v}}, \mathbf{t}'), \mathbf{t} \neq \mathbf{t}'. \end{aligned}$$

$$\begin{aligned}\varphi_3' : \texttt{other}_R(\mathbf{x},\mathbf{s},\mathbf{y},\bar{\mathbf{v}},\mathbf{t}') \quad\leftarrow\quad & \mathit{pos}(\varphi_2')|_{sch(P)}, \mathit{neg}(\varphi_2'), \mathit{neq}(\varphi_2'), \texttt{all}(\mathbf{y}),\\ & \texttt{clock}(\mathbf{x},\mathbf{s},\mathbf{y},\mathbf{u}), \texttt{time}(\mathbf{t}'), \mathbf{u}\le\mathbf{t}',\\ & \texttt{chosen}_R(\mathbf{x},\mathbf{s},\mathbf{y},\bar{\mathbf{v}},\mathbf{t}), \mathbf{t}'\neq\mathbf{t}.\end{aligned}$$

By the correspondence of ψ_3 to φ_2 we have a valuation V for φ_2, such that $\mathbf{C}$ is the sequence consisting of $V(\mathit{pos}(\varphi_2))$ and $V(\mathit{neq}(\varphi_2))$. The valuation V can be extended to a valuation for φ_3 by adding the mappings $\mathbf{u}\mapsto u$ and $\mathbf{t}'\mapsto t'$. This goes similarly for ψ_4, φ_2' and $\mathbf{C}'$. Therefore the following ground rules are in $G_M(P)$:

$$\begin{aligned}\psi_5 : \texttt{other}(x,s,y,\bar{a},t) \quad\leftarrow\quad & \mathbf{C}, \texttt{all}(y),\\ & \texttt{clock}(x,s,y,u), \texttt{time}(t), u\le t,\\ & \texttt{chosen}_R(x,s,y,\bar{a},t'), t\neq t'.\end{aligned}$$

$$\begin{aligned}\psi_6 : \texttt{other}(x,s,y,\bar{a},t') \quad\leftarrow\quad & \mathbf{C}', \texttt{all}(y),\\ & \texttt{clock}(x,s,y,u), \texttt{time}(t'), u\le t'\\ & \texttt{chosen}_R(x,s,y,\bar{a},t), t'\neq t.\end{aligned}$$

Since the bodies of ψ_3 and ψ_4 are true on M, and because $\texttt{chosen}_R(x,s,y,\bar{a},t)\in M$, $\texttt{chosen}_R(x,s,y,\bar{a},t')\in M$ and $t\neq t'$, we obtain that the bodies of ψ_5 and ψ_6 are also true on M. Now, since M is a fixpoint, we must have $\texttt{other}_R(x,s,y,\bar{a},t)\in M$ and $\texttt{other}_R(x,s,y,\bar{a},t')\in M$, and we have arrived at the contradiction. □

Lemma F.10. *Let $\psi\in G_M(P)$ be a ground rule of the form (4.8):*

$$\begin{aligned}R_{\text{snd}}(x,s,y,t,\bar{a}) \quad\leftarrow\quad & \mathbf{B}, \texttt{all}(y), \texttt{clock}(x,s,y,u),\\ & \texttt{time}(t), u\le t,\\ & \texttt{chosen}_R(x,s,y,\bar{a},t).\end{aligned}$$

where $\mathbf{B}$ is a sequence of positive ground atoms (facts) and ground nonequalities. Suppose that $M\models\mathbf{B}$, $\texttt{all}(y)\in \mathit{decl}(H)\subseteq M$, $\texttt{clock}(x,s,y,u)\in M$, $\texttt{time}(t)\in\mathit{decl}(H)\subseteq M$ and $u\le t$.

Now, if $\texttt{chosen}_R(x,s,y,\bar{a},t)\notin M$ then there exists a value $t'\in\mathbb{N}$ with $u\le t'$ such that $\texttt{chosen}_R(x,s,y,\bar{a},t')\in M$.

Proof. Assume the contrary. Consider a value $t'\in\mathbb{N}$ such that $u\le t'$ (and thus $\texttt{time}(t')\in\mathit{decl}(H)\subseteq M$). Consider the following ground rule, based on the above ground rule except that t has been replaced by t':

$$\begin{aligned}\psi : R_{\text{snd}}(x,s,y,t',\bar{a}) \quad\leftarrow\quad & \mathbf{B}, \texttt{all}(y), \texttt{clock}(x,s,y,u),\\ & \texttt{time}(t'), u\le t',\\ & \texttt{chosen}_R(x,s,y,\bar{a},t')\end{aligned}$$

By the given, $M\models\mathbf{B}$, $\texttt{all}(y)\in\mathit{decl}(H)\subseteq M$ and $\texttt{clock}(x,s,y,u)\in M$. By contradiction we have assumed that $\texttt{chosen}_R(x,s,y,\bar{a},t')\notin M$. The rule ψ is based on a non-ground rule φ in $\mathit{pure}(P)$ and a valuation V for φ, which is of the form (4.8). In turn, φ is based on an original asynchronous rule φ_2 in P such that $\mathbf{B}$ is the sequence consisting of $V(\mathit{pos}(\varphi_2))$ and $V(\mathit{neq}(\varphi_2))$. Based on φ_2 there is a rule φ_3 in $\mathit{pure}(P)$ of the form (4.9), that derives $\texttt{chosen}_R$-facts. Although the fact $\texttt{chosen}_R(x,s,y,\bar{a},t')$ could in principle be derived by multiple ground rules, $\texttt{chosen}_R(x,s,y,\bar{a},t')\notin M$ implies that amongst others the following ground rule based on φ_3 and V is absent from $G_M(P)$:

$$\begin{aligned}\texttt{chosen}_R(x,s,y,\bar{a},t') \quad\leftarrow\quad & \mathbf{B}, \texttt{all}(y), \texttt{clock}(x,s,y,u),\\ & \texttt{time}(t'), u\le t'.\end{aligned}$$

Looking at the form (4.9), the absence of the previous ground rule from $G_M(P)$ implies that $\texttt{other}_R(x,s,y,\bar{a},t')\in M$. This $\texttt{other}_R$-fact must be derived by a ground rule, of the following form, based on (4.10):

$$\begin{aligned}\texttt{other}_R(x,s,y,\bar{a},t') \quad\leftarrow\quad & \mathbf{B}', \texttt{all}(y), \texttt{clock}(x,s,y,v),\\ & \texttt{time}(t'), v\le t',\\ & \texttt{chosen}_R(x,s,y,\bar{a},t''), t''\neq t'.\end{aligned}$$

Thus apparently there is some value $t''\in\mathbb{N}$ such that $\texttt{chosen}_R(x,s,y,\bar{a},t'')\in M$. Again, the fact $\texttt{chosen}_R(x,s,y,\bar{a},t'')$ was derived by a ground rule of the following form, based on (4.9):

$$\begin{aligned}\texttt{chosen}_R(x,s,y,\bar{a},t'') \quad\leftarrow\quad & \mathbf{B}'', \texttt{all}(y), \texttt{clock}(x,s,y,w),\\ & \texttt{time}(t''), w\le t''.\end{aligned}$$

Using Lemma F.3 and $\texttt{clock}(x,s,y,u)\in M$, we know that $w=u$ and therefore $u\le t''$. Thus there exists $t''\in\mathbb{N}$ with $u\le t''$ and $\texttt{chosen}_R(x,s,y,\bar{a},t'')\in M$, and we have arrived at the contradiction. □

F.3 Local vector clocks

Based on Corollary F.8, for $x \in \mathcal{N}$ and $s \in \mathbb{N}$ we define the *(local) vector clock* associated with node x at local timestamp s, denoted $v_M(x,s)$, as follows: for $y \in \mathcal{N}$, we set $v_M(x,s)[y] = t$ such that $\texttt{clock}(x,s,y,t) \in M$.

The following lemmas provide insight into these local vector clocks.

Lemma F.11. *Let $x, y \in \mathcal{N}$ and $s \in \mathbb{N}$. A fact $\texttt{rcvClock}(x,s,y,u) \in M$ implies $v_M(x,s)[y] \geq u$.*

Proof. Denote $v_M(x,s)[y] = v$. Suppose by contradiction that $v < u$. Then by definition of $v_M(x,s)[y]$, we have $\texttt{clock}(x,s,y,v) \in M$, and thus $\texttt{rcvClock}(x,s,x,v) \in M$ by the rule (4.7). Because both $\texttt{rcvClock}(x,s,y,u) \in M$ and $\texttt{rcvClock}(x,s,y,v) \in M$, a ground rule of the form (4.6) derives $\texttt{isBehind}(x,s,y,v) \in M$ because M is a fixpoint. Consider the following ground rule, based on rule (4.7):

$$\texttt{clock}(x,s,y,v) \leftarrow \texttt{rcvClock}(x,s,y,v).$$

Because $\texttt{isBehind}(x,s,y,v) \in M$, this rule is not in $G_M(P)$ and because there are no other ground rules with this head, we have $\texttt{clock}(x,s,y,v) \notin M$, a contradiction. □

Lemma F.12. *Let $x \in \mathcal{N}$ and $s \in \mathbb{N}$. We have $v_M(x,s)[x] = s+1$.*

Proof. First we show that $v_M(x,s)[x] \geq s+1$. We have $\texttt{rcvClock}(x,s,x,s+1) \in M$ by Lemma F.2. Then by Lemma F.11 we have $v_M(x,s)[x] \geq s+1$.

Now we show that $v_M(x,s)[x] \leq s+1$, which combined with the above implies $v_M(x,s)[x] = s+1$. As a proof by contradiction, suppose that $v_M(x,s)[x] > s+1$. Abbreviate $u = v_M(x,s)[x]$. By definition of u, we have $\texttt{clock}(x,s,x,u) \in M$ and thus $\texttt{rcvClock}(x,s,x,u) \in M$ by the rule (4.7). Because $u > s+1$, the fact $\texttt{rcvClock}(x,s,x,u)$ can not have been derived by ground rules of the forms (4.3), (4.4) and (4.5): ground rules of the forms (4.3) or (4.5) would require the first and third components to be different, whereas ground rules of the form (4.4) would force $u = s+1$. So, $\texttt{rcvClock}(x,s,x,u) \in M$ can only have been derived by means of a ground rule of the form (4.12):

$$\texttt{rcvClock}(x,s,x,u) \leftarrow R_{\text{snd}}(y,t,x,s,\bar{a}), \texttt{clock}(y,t,x,u).$$

This implies $R_{\text{snd}}(y,t,x,s,\bar{a}) \in M$ and $\texttt{clock}(y,t,x,u) \in M$. The fact $R_{\text{snd}}(y,t,x,s,\bar{a}) \in M$ was derived by a ground rule of the form (4.8):

$$\begin{aligned} R_{\text{snd}}(y,t,x,s,\bar{a}) \quad \leftarrow \quad & \mathbf{B}, \texttt{all}(x), \texttt{clock}(y,t,x,w), \\ & \texttt{time}(s), w \leq s, \\ & \texttt{chosen}_R(y,t,x,\bar{a},s). \end{aligned}$$

Using Corollary F.8 and $\texttt{clock}(y,t,x,u) \in M$ from above, we find that $w = u$. Since the body of the previous ground rule is true, thus $u \leq s$ and $u < s+1$. We have arrived at the contradiction. □

Lemma F.13. *Let $x \in \mathcal{N}$ and $s \in \mathbb{N}$. We have $v_M(x,s) \prec v_M(x,s+1)$.*

Proof. We first show for $y \in \mathcal{N}$ with $y \neq x$ that $v_M(x,s)[y] \leq v_M(x,s+1)[y]$. Let $u = v_M(x,s)[y]$. By definition of $v_M(x,s)[y]$, we have $\texttt{clock}(x,s,y,u) \in M$. The following ground rule, based on (4.5), is in $G_M(P)$ and it derives $\texttt{rcvClock}(x,s+1,y,u) \in M$:

$$\begin{aligned} \texttt{rcvClock}(x,s+1,y,u) \quad \leftarrow \quad & \texttt{clock}(x,s,y,u), x \neq y, \\ & \texttt{tsucc}(s,s+1). \end{aligned}$$

Then by Lemma F.11 we have $v_M(x,s+1)[y] \geq u$ as desired.

By Lemma F.12 we have $v_M(x,s)[x] = s+1 < s+2 = v_M(x,s+1)[x]$ and together with the above we conclude $v_M(x,s) \prec v_M(x,s+1)$. □

Lemma F.14. *For each $R_{\text{snd}}(x,s,y,t,\bar{a}) \in M$ with R a relation in sch(P) we have $v_M(x,s) \prec v_M(y,t)$.*

Proof. First we show that $v_M(x,s) \preceq v_M(y,t)$. Let $z \in \mathcal{N}$. We must show that $v_M(x,s)[z] \leq v_M(y,t)[z]$. Abbreviate $u = v_M(x,s)[z]$. By definition of $v_M(x,s)[z]$ we have $\texttt{clock}(x,s,z,u) \in M$. The following rule derives $\texttt{rcvClock}(y,t,z,u) \in M$ because M is a fixpoint:

$$\texttt{rcvClock}(y,t,z,u) \leftarrow R_{\text{snd}}(x,s,y,t,\bar{a}), \texttt{clock}(x,s,z,u).$$

By Lemma F.11 we then have $u \leq v_M(y,t)[z]$. Because $z \in \mathcal{N}$ was taken in general, we have $v_M(x,s) \preceq v_M(y,t)$.

Now we show that $v_M(x,s)[y] < v_M(y,t)[y]$. The fact $R_{\text{snd}}(x,s,y,t,\bar{a})$ was derived by a ground rule of the form (4.8):

$$\begin{aligned} R_{\text{snd}}(x,s,y,t,\bar{a}) \quad \leftarrow \quad & \mathbf{B}, \texttt{all}(y), \\ & \texttt{clock}(x,s,y,w), \texttt{time}(t), w \leq t, \\ & \texttt{chosen}_R(x,s,y,\bar{a},t). \end{aligned}$$

Here $\mathbf{B}$ is a sequence of positive ground atoms (facts) and ground nonequalities. Because the body of this rule is true on M, we have $\texttt{clock}(x, s, y, w) \in M$. By definition of $v_M(x, s)$, we have $v_M(x, s)[y] = w \leq t$. By Lemma F.12 we have $v_M(y, t)[y] = t+1$ and thus $v_M(x, s)[y] < v_M(y, t)[y]$.

Now combining $v_M(x, s) \preceq v_M(y, t)$ and $v_M(x, s)[y] < v_M(y, t)[y]$ gives us $v_M(x, s) \prec v_M(y, t)$. □

Lemma F.15. *Let $(x, s) \in \mathcal{N} \times \mathbb{N}$ and $(y, t) \in \mathcal{N} \times \mathbb{N}$. If $(x, s) \neq (y, t)$ then $v_M(x, s) \neq v_M(y, t)$.*

Proof. Let $(x, s) \in \mathcal{N} \times \mathbb{N}$ and $(y, t) \in \mathcal{N} \times \mathbb{N}$ be such that $(x, s) \neq (y, t)$. We show that $v_M(x, s) \neq v_M(y, t)$. First suppose that $x = y$ but $s \neq t$. Suppose without loss of generalisation that $s < t$. Then $v_M(x, s) \prec v_M(x, t)$ by Lemma F.13 and thus $v_M(x, s) \neq v_M(y, t)$.

Now suppose that $x \neq y$. We show that $v_M(x, s) \neq v_M(y, t)$. As a proof by contradiction, suppose that $v_M(x, s) = v_M(y, t)$. Abbreviate $u = v_M(x, s)[x]$. Since $v_M(x, s) = v_M(y, t)$ by assumption, we have $v_M(y, t)[x] = u$. By definition of $v_M(y, t)$ we have $\texttt{clock}(y, t, x, u) \in M$. Intuitively, we will now trace the derivation of this $\texttt{clock}$-fact backwards, and arrive at a contradiction.

By definition of $u = v_M(x, s)[x]$, we have $u = s + 1$ by Lemma F.12 and thus $u > 0$. The fact $\texttt{clock}(y, t, x, u) \in M$ implies $\texttt{rcvClock}(y, t, x, u) \in M$, by the rule (4.7). This $\texttt{rcvClock}$-fact cannot be generated by ground rules of the forms (4.3) and (4.4), because they force $u = 0$ and $y = x$ respectively. The fact $\texttt{rcvClock}(y, t, x, u)$ can thus have been generated only by ground rules of the forms (4.5) or (4.12). We will name these forms the *local-rule* and *send-rule* respectively. First, if the local-rule applies to the origin of $\texttt{rcvClock}(y, t, x, u)$, then $\texttt{clock}(y, t - 1, x, u) \in M$ and thus $\texttt{rcvClock}(y, t - 1, x, u) \in M$, by the rule (4.7). We can repeat the same backward reasoning with the local-rule until we hit a local time $t' \leq t$ of y such that $\texttt{clock}(y, t', x, u) \in M$ but $\texttt{clock}(y, t' - 1, x, u) \notin M$. At that moment only the send-rule can be applied. Thus, $\texttt{rcvClock}(y, t', x, u) \in M$ was generated by a ground rule of the form (4.12):

$$\texttt{rcvClock}(y, t', x, u) \quad \leftarrow \quad R_{\text{snd}}(z, v, y, t', \bar{a}), \texttt{clock}(z, v, x, u).$$

Thus $R_{\text{snd}}(z, v, y, t', \bar{a}) \in M$, $\texttt{clock}(z, v, x, u) \in M$. Now we follow this send-rule backward and jump to the node z. Now again, we use the local-rule on node z until we hit a local time $v' \leq v$ of z such that $\texttt{clock}(z, v', x, u) \in M$ but $\texttt{clock}(z, v' - 1, x, u) \notin M$. Then we are forced to follow a send-rule backward again.

We can keep repeating this reasoning: when we arrive at a node by means of a send-rule, then we first try to follow as many local-rules as possible until we are forced to use a send-rule again. This gives us a sequence $Z = (z_0, l_0), (z_1, l_1), (z_2, l_2), \ldots$ such that for $i = 0, 1, 2, \ldots$ we have $\texttt{clock}(z_i, l_i, x, u) \in M$ but $\texttt{clock}(z_i, l_i - 1, x, u) \notin M$ and thus we jumped from node z_i to node z_{i+1} by means of a send-rule. We have $z_0 = y$ and $l_0 \leq t$. At this point we don't know whether the sequence Z is finite.

We now show that we eventually end up following a send-rule backward to the node x itself. By Lemma F.13 (for the local-rule) and Lemma F.14 (for the send-rule), we obtain for each index i of Z for which the index $i + 1$ is also defined that $v_M(z_i, l_i) \succ v_M(z_{i+1}, l_{i+1})$. So intuitively, the sequence Z respects a (reversed) causal order. Now we show that nodes are not repeated in the sequence Z. Suppose that there is a repetition of a node z in Z: there are indices i and j with $0 \leq i < j$ such that $z_i = z_j = z$. Using the previously mentioned causal ordering of the sequence Z we have $v_M(z, l_i) \succ v_M(z, l_j)$. Using Lemma F.13 this implies $l_j < l_i$. By construction of sequence Z we have $\texttt{clock}(z, l_i, x, u) \in M$ and $\texttt{clock}(z, l_j, x, u) \in M$. Then by definition of the local vector clocks, we have $v_M(z, l_j)[x] = u$ and $v_M(z, l_i)[x] = u$. Now take l such that $l_j + 1 \leq l \leq l_i - 1$. We have $v_M(z, l_j) \preceq v_M(z, l) \preceq v_M(z, l_i)$ by Lemma F.13. Thus $u = v_M(z, l_j)[x] \leq v_M(z, l)[x] \leq v_M(z, l_i)[x] = u$ and we thus find that $v_M(z, l)[x] = u$. For the case $l = l_i - 1$, this implies $\texttt{clock}(z_i, l_i - 1, x, u) \in M$, but this is a contradiction. So, there can be no repetition of a node in the sequence Z. This implies that Z is finite because $\mathcal{N}$ is finite and because M is well-formed. Also, Z contains x because while we are building the sequence Z, on a node $z \neq x$ we are forced to (eventually) follow a send-rule to another node that does not yet occur in the sequence.

We have shown that we eventually follow a send-rule back to node x. This rule is of the form (4.12):

$$\texttt{rcvClock}(z, v, x, u) \leftarrow R_{\text{snd}}(x, s', z, v, \bar{a}), \texttt{clock}(x, s', x, u).$$

Since $\texttt{clock}(x, s', x, u) \in M$, we have defined $v_M(x, s')[x] = u$. Now there are three options that each result in a contradiction:

1. Suppose $s' = s$. By Lemma F.14 we then have $v_M(x, s) \prec v_M(z, v)$. If $(z, v) = (y, t)$ then immediately $v_M(x, s) \neq v_M(y, t)$, a contradiction. Suppose $(z, v) \neq (y, t)$. Then by construction of the sequence Z we have $v_M(x, s) \prec v_M(z, v) \preceq v_M(z_0, l_0) \preceq v_M(y, t)$ and because $\preceq$ is partial order on vector clocks, therefore $v_M(x, s) \prec v_M(y, t)$ and thus $v_M(x, s) \neq v_M(y, t)$, a contradiction.
2. Suppose $s' < s$. By Lemma F.12 we have $v_M(x, s')[x] = s' + 1 < s + 1 = v_M(x, s)[x]$, which implies $u < u$, a contradiction.
3. Suppose $s < s'$. Similarly to the previous case, by Lemma F.12 we must have $v_M(x, s)[x] = s + 1 < s' + 1 = v_M(x, s')[x]$, which again implies $u < u$, a contradiction.

Therefore, if $x \neq y$, supposing that $v_M(x, s) = v_M(y, t)$ leads to a contradiction. Therefore $v_M(x, s) \neq v_M(y, t)$.

Thus overall, $(x, s) \neq (y, t)$ implies $v_M(x, s) \neq v_M(y, t)$. □

F.4 Construction of run

Consider the definitions in Section 6.2. We first show that relation $\preceq_L$ on L is actually a partial order. This relation is reflexive and transitive because the partial order $\preceq$ on vector clocks is reflexive and transitive. We now show that it is antisymmetric as well. Let $(x, s) \in L$ and $(y, t) \in L$ be such that $(x, s) \preceq_L (y, t)$ and $(y, t) \preceq_L (x, s)$. Then $v_M(x, s) \preceq v_M(y, t)$

and $v_M(y,t) \preceq v_M(x,s)$ by definition of $\preceq_L$. Because $\preceq$ is a partial order on vector clocks, we have $v_M(x,s) = v_M(y,t)$ and, by Lemma F.15 this implies $(x,s) = (y,t)$, as desired. Therefore, $\preceq_L$ is a partial order on L.

We will henceforth abbreviate the partial order $\preceq_L$ and the total order $\leq_L$ as $\preceq$ and $\leq$ respectively. It will be clear from the context whether "$\preceq$" denotes the partial order on L or the partial order on vector clocks.

As a converse to the function *ord*, let $p : \mathbb{N} \to L$ be the function that maps an ordinal i to the pair (x,s) on position i of C. Note that $p(\mathit{ord}(x,s)) = (x,s)$ and $\mathit{ord}(p(i)) = i$ for $(x,s) \in L$ and $i \in \mathbb{N}$.

F.4.1 Properties

The following properties are relative to the sequence $\mathcal{R}$, as defined in Section (6.2.2).

Lemma F.16. *The sequence $\mathcal{R}$ constructed from M is a fair run of P on input H and the trace of $\mathcal{R}$ is M.*

Proof. Using Lemma F.17 we know that $\rho_0 = \mathit{start}(P,H)$. Now, for each tuple $i \in \mathbb{N}$ of $\mathcal{R}$, apply Lemma F.18 to see that tuple i is a valid global transition, using send-tag i. Furthermore, the ending-configuration of one global transition is the begin-configuration of the next one. Therefore $\mathcal{R}$ is a valid run of P on input H.

Now we show that $\mathcal{R}$ is fair. Let $i \in \mathbb{N}$. Firstly, every node $x \in \mathcal{N}$ is the recipient in an infinite number of global transitions because there are an infinite number of pairs in L with first component x. Now we show that every message is eventually delivered. Let $\langle k, \boldsymbol{f}\rangle \in b^{\rho_i}$ with $\boldsymbol{f}$ of the form $R(y,\bar{a})$. By definition of b^{ρ_i}, $\langle k, \boldsymbol{f}\rangle \in b^{\rho_i}$ means that there is some fact $R_{\mathrm{snd}}(x,s,y,t,\bar{a}) \in M$ such that $k = \mathit{ord}(x,s) < i \leq \mathit{ord}(y,t)$. Denote $j = \mathit{ord}(y,t)$, so $i \leq j$. By definition, $m_j = \mathit{pairs}(\mathrm{deliv}_j)$ with

$$\mathrm{deliv}_j = \{R_{\mathrm{snd}}(x',s',y',t',\bar{b}) \in M \mid \mathit{ord}(y',t') = j\}.$$

Therefore, $R_{\mathrm{snd}}(x,s,y,t,\bar{a}) \in \mathrm{deliv}_j$ and $\langle k, \boldsymbol{f}\rangle \in m_j$. We conclude that the run $\mathcal{R}$ is fair.

Finally, the trace of $\mathcal{R}$ is M by Lemma F.21. □

Lemma F.17. *We have $\rho_0 = \mathit{start}(P,H)$.*

Proof. Abbreviate $\rho = \mathit{start}(P,H)$. First we show that $s^{\rho_0} = s^{\rho}$. First, we have $\mathit{local}_C(0,x) = |\{(x,s) \in L \mid \mathit{ord}(x,s) < 0\}|$. But there is no fact $(x,s) \in L$ with $\mathit{ord}(x,s) < 0$, so $\mathit{local}_C(0,x) = 0$. Now we have the following:

$$\begin{aligned} s^{\rho_0} &= \bigcup_{x\in\mathcal{N}} \mathit{state}(x, \mathit{local}_C(0,x)) && \text{(by definition)} \\ &= \bigcup_{x\in\mathcal{N}} \mathit{state}(x,0) \\ &= \bigcup_{x\in\mathcal{N}} \big(H|^x \cup (M^{\mathrm{ind}}|^{x,0})^{\downarrow}\big). && \text{(by definition)} \end{aligned}$$

The timestamp in the head of inductive ground rules whose body is true on M must be at least 1 (this is enforced by the ground `tsucc`-atom). Thus for any x we have $M^{\mathrm{ind}}|^{x,0} = \emptyset$. Therefore, $s^{\rho_0} = \bigcup_{x\in\mathcal{N}} H|^x = s^{\rho}$.

Now we show that $b^{\rho_0} = b^{\rho}$. By definition $b^{\rho_0} = \mathit{pairs}(\mathrm{buf}_0)$ with

$$\mathrm{buf}_0 = \{R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M \mid \mathit{ord}(y,t) < 0 \leq \mathit{ord}(z,u)\}.$$

But $\mathrm{buf}_0 = \emptyset$ because there are no pairs $(y,t) \in L$ such that $\mathit{ord}(y,t) < 0$. Therefore $b^{\rho_0} = \emptyset = b^{\rho}$. □

Lemma F.18. *Let $i \in \mathbb{N}$. We have that $\rho_i \xrightarrow[i]{x_i, m_i} \rho_{i+1}$ is a valid global transition of P on input H.*

Proof. We first show that $m_i \subseteq b^{\rho_i}$. By Lemma F.14, for a fact $R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M$ we have $v_M(y,t) \prec v_M(z,u)$. From this it follows that $\mathit{ord}(y,t) < \mathit{ord}(z,u)$. We now have

$$\begin{aligned} \mathrm{deliv}_i &= \{R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M \mid \mathit{ord}(z,u) = i\} && \text{(by definition)} \\ &= \{R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M \mid \mathit{ord}(y,t) < \mathit{ord}(z,u) = i\} && \text{(see above)} \\ &\subseteq \{R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M \mid \mathit{ord}(y,t) < i \leq \mathit{ord}(z,u)\} \\ &= \mathrm{buf}_i. && \text{(by definition)} \end{aligned}$$

Therefore $\mathit{pairs}(\mathrm{deliv}_i) \subseteq \mathit{pairs}(\mathrm{buf}_i)$ and thus $m_i \subseteq b^{\rho_i}$.

Now we show that all facts in m_i have location specifier x_i. Consider a pair $\langle k, R(z,\bar{a})\rangle \in m_i$. By definition of m_i, there is a fact $R_{\mathrm{snd}}(y,t,z,u,\bar{a}) \in M$ with $\mathit{ord}(z,u) = i$. Thus $p(i) = (z,u)$ and $z = x_i$ by definition of x_i. Thus all facts in m_i have location specifier x_i.

Because $m_i \subseteq b^{\rho_i}$ holds, we can consider the unique result configuration ρ such that $\rho_i \xrightarrow[i]{x_i, m_i} \rho$ is a valid global transition using send-tag i. We have to show that $\rho_{i+1} = \rho$. We do this in two parts: *(i)* $s^{\rho_{i+1}} = s^{\rho}$ (state) and *(ii)* $b^{\rho_{i+1}} = b^{\rho}$ (buffers).

Denote $S = s^{\rho_i}|^{x_i} \cup \mathit{untag}(m_i)$ and $D = \mathit{deduc}(P)(S)$.

State.

Denote $I = \mathit{induc}(P)(D)$. Recall that by definition,

$$s^{\rho_i} = \bigcup_{z\in\mathcal{N}} \mathit{state}(z, \mathit{local}_C(i, z)),$$

$$s^{\rho_{i+1}} = \bigcup_{z\in\mathcal{N}} \mathit{state}(z, \mathit{local}_C(i+1, z)).$$

To show $s^{\rho_{i+1}} = s^{\rho}$ we must show both inclusions $s^{\rho_{i+1}} \subseteq s^{\rho}$ and $s^{\rho} \subseteq s^{\rho_{i+1}}$.

Let $\boldsymbol{f} \in s^{\rho_{i+1}}$. We show that $\boldsymbol{f} \in s^{\rho}$. Let y be the location specifier in $\boldsymbol{f}$. Denote $s = \mathit{local}_C(i, y)$ and $t = \mathit{local}_C(i+1, y)$. We have $\boldsymbol{f} \in \mathit{state}(y, t) = H|^{y} \cup (M^{\mathrm{ind}}|^{y,t})^{\downarrow}$.

- Suppose $y \neq x_i$. Then $\mathit{local}_C(i, y) = \mathit{local}_C(i+1, y)$. Thus $t = s$ and $\boldsymbol{f} \in H|^{y} \cup (M^{\mathrm{ind}}|^{y,s})^{\downarrow} = \mathit{state}(y, s) \subseteq s^{\rho_i}$, by definition of ρ_i. Furthermore, because y is not the recipient in the global transition $\rho_i \xrightarrow[i]{x_i, m_i} \rho$, by the semantics of global transitions we obtain $\boldsymbol{f} \in s^{\rho}$.
- Suppose $y = x_i$. First, suppose that the predicate of $\boldsymbol{f}$ is in $\mathit{edb}(P)^{\mathrm{L}}$. In that case $\boldsymbol{f} \in H|^{y} \subseteq s^{\rho_i}$. By the semantics of local transitions (that preserve EDB facts) we then obtain $\boldsymbol{f} \in s^{\rho}$.

  Now suppose that the predicate of $\boldsymbol{f}$ is not in $\mathit{edb}(P)^{\mathrm{L}}$. By definition of $s^{\rho_{i+1}}$ we must have $\boldsymbol{f} \in (M^{\mathrm{ind}}|^{y,t})^{\downarrow}$ and thus $\boldsymbol{f}^{\uparrow t} \in M^{\mathrm{ind}}|^{y,t}$. We have $t = s+1$ because $y = x_i$. Because $\boldsymbol{f}^{\uparrow t} \in M^{\mathrm{ind}}|^{y,t}$, there must be a ground inductive rule ψ in $G_M(P)$ with head $\boldsymbol{f}^{\uparrow t}$ whose ground nonequalities are true and such that $\mathit{pos}(\psi)|_{\mathit{sch}(P)} \subseteq M|^{x_i,s}$, because the timestamp in the head is the successor of the timestamp in the body. Furthermore, because ψ exists in $G_M(P)$, it is possible to choose a non-ground inductive rule φ from $\mathit{pure}(P)$ and a valuation V of φ such that ψ is the ground version of φ based on V (without negative body literals) and such that $V(\mathit{pos}(\varphi)|_{\mathit{sch}(P)}) \subseteq M|^{x_i,s}$ and $V(\mathit{neg}(\varphi)) \cap M = \emptyset$. Now because $M|^{x_i,s} \subseteq M$, we have $V(\mathit{neg}(\varphi)) \cap M|^{x_i,s} = \emptyset$. Let φ' denote the simplified version of φ, which is in $\mathit{induc}(P)$. Let V' denote the restriction of V to the variables of φ'. We have $V'(\mathit{head}(\varphi')) = \boldsymbol{f}$. By Lemma F.19 we have $D^{\uparrow s} = M|^{x_i,s}$. Thus $V(\mathit{pos}(\varphi)|_{\mathit{sch}(P)}) \subseteq M|^{x_i,s}$ implies $V'(\mathit{pos}(\varphi')) \subseteq D$ and $V(\mathit{neg}(\varphi)) \cap M|^{x_i,s} = \emptyset$ implies $V'(\mathit{neg}(\varphi')) \cap D = \emptyset$. Therefore V' is satisfying for the simplified version φ' on input D and thus we derive the fact $\boldsymbol{f} \in I \subseteq s^{\rho}$.

Let $\boldsymbol{f} \in s^{\rho}$. We show that $\boldsymbol{f} \in s^{\rho_{i+1}}$. Let y be the location specifier in $\boldsymbol{f}$. Denote again $s = \mathit{local}_C(i, y)$ and $t = \mathit{local}_C(i+1, y)$.

- Suppose $y \neq x_i$. This implies $t = s$. By the semantics of global transitions, we have $\boldsymbol{f} \in s^{\rho_i}$ because during the global transition, facts without location specifier x_i are copied unmodified to s^{ρ} and new facts must have location specifier x_i. By definition of s^{ρ_i} we must have $\boldsymbol{f} \in \mathit{state}(y, s)$. But since $s = t$ we have by definition of $s^{\rho_{i+1}}$ that $\boldsymbol{f} \in \mathit{state}(y, t) \subseteq s^{\rho_{i+1}}$.
- Suppose $y = x_i$. This implies $t = s+1$. First, suppose that the predicate of $\boldsymbol{f}$ is in $\mathit{edb}(P)^{\mathrm{L}}$. In that case $\boldsymbol{f} \in s^{\rho_i}$ by using the semantics of local transitions (that preserve EDB facts). Now, by definition of s^{ρ_i} we then have $\boldsymbol{f} \in H|^{y}$. Thus by definition of $s^{\rho_{i+1}}$ also $\boldsymbol{f} \in \mathit{state}(y, t) \subseteq s^{\rho_{i+1}}$.

  Now suppose that the predicate of $\boldsymbol{f}$ is not in $\mathit{edb}(P)^{\mathrm{L}}$. Then there must be a simplified inductive rule φ in $\mathit{induc}(P)$ and valuation V of φ that have produced $\boldsymbol{f}$ during the global transition $\rho_i \xrightarrow[i]{x_i, m_i} \rho$, thus $\boldsymbol{f} \in \mathit{induc}(P)(D)$. Let φ' be the original inductive rule of P on which φ is based. We can extend V to a valuation V' for φ' to assign value s to the body timestamp-variable and $t = s+1$ to the head timestamp-variable. We have $V(\mathit{pos}(\varphi)) \subseteq D$ and $V(\mathit{neg}(\varphi)) \cap D = \emptyset$. Therefore $V'(\mathit{pos}(\varphi')) \subseteq D^{\uparrow s}$ and $V'(\mathit{neg}(\varphi')) \cap D^{\uparrow s} = \emptyset$. By Lemma F.19 we have $D^{\uparrow s} = M|^{x,s}$ and thus $V'(\mathit{pos}(\varphi')) \subseteq M|^{x,s} \subseteq M$ and $V'(\mathit{neg}(\varphi')) \cap M|^{x,s} = \emptyset$. The latter implies $V'(\mathit{neg}(\varphi')) \cap M = \emptyset$ because all facts in $V'(\mathit{neg}(\varphi'))$ have location specifier x and timestamp s. Therefore, the positive ground rule ψ based on φ' and V' (without negative body literals) is in $G_M(P)$ and it derives $\boldsymbol{f}^{\uparrow t} \in M^{\mathrm{ind}}|^{x,t}$ because M is a fixpoint. By definition of $s^{\rho_{i+1}}$ we then have $\boldsymbol{f} \in \mathit{state}(y, t) \subseteq s^{\rho_{i+1}}$.

Buffers .

Denote $\delta = \mathit{async}(P)(D)$. Recall the notation $\mathit{tag}(i, \delta) = \{\langle i, \boldsymbol{g}\rangle \mid \boldsymbol{g} \in \delta\}$.

By the semantics of global transitions, we have $b^{\rho} = (b^{\rho_i} \setminus m_i) \cup \mathit{tag}(i, \delta)$. Denote $s = \mathit{local}_C(i, x_i)$. By Lemma F.23 we have $\mathit{ord}(x_i, s) = i$.

Let $\langle k, \boldsymbol{f}\rangle \in b^{\rho_{i+1}}$. We show that $\langle k, \boldsymbol{f}\rangle \in b^{\rho}$. Fact $\boldsymbol{f}$ is of the form $R(z, \bar{a})$. By definition of $b^{\rho_{i+1}}$ there must be a fact $R_{\mathrm{snd}}(y, t, z, u, \bar{a}) \in M$ with $k = \mathit{ord}(y, t) < i+1 \leq \mathit{ord}(z, u)$. Thus $i < \mathit{ord}(z, u)$.

- Suppose that $(y, t) \neq (x_i, s)$. Then $\mathit{ord}(y, t) \neq \mathit{ord}(x_i, s)$ because ord is injective. Since $\mathit{ord}(x_i, s) = i$, this implies $\mathit{ord}(y, t) \neq i$ and combined with $\mathit{ord}(y, t) < i+1$ we thus have $\mathit{ord}(y, t) < i$. In that case $\langle k, \boldsymbol{f}\rangle \in b^{\rho_i}$ by definition of b^{ρ_i}. We now show that $\langle k, \boldsymbol{f}\rangle \notin m_i$, which would give $\langle k, \boldsymbol{f}\rangle \in (b^{\rho_i} \setminus m_i) \subseteq b^{\rho}$. First, because $i < \mathit{ord}(z, u)$ we have $R_{\mathrm{snd}}(y, t, z, u, \bar{a}) \notin \mathrm{deliv}_i$. Also, there can be no fact $R_{\mathrm{snd}}(y, t, z, v, \bar{a}) \in \mathrm{deliv}_i$ with $v \neq u$ by Lemma F.9. Additionally, there is no $(y', t') \in L$ with $(y', t') \neq (y, t)$ and $\mathit{ord}(y', t') = k$ because ord is injective. Thus $\langle k, \boldsymbol{f}\rangle \notin m_i$.
- Suppose that $(y, t) = (x_i, s)$. Then $k = i$. We show that $\langle k, \boldsymbol{f}\rangle \in \mathit{tag}(i, \delta)$, which would give $\langle k, \boldsymbol{f}\rangle \in b^{\rho}$. Because $(y, t) = (x_i, s)$ we have $R_{\mathrm{snd}}(y, t, z, u, \bar{a}) = R_{\mathrm{snd}}(x_i, s, z, u, \bar{a}) \in M$. This fact must be generated by a ground rule

$\psi \in G_M(P)$ of the form (4.8):

$$\begin{aligned} R_{\text{snd}}(x_i, s, z, u, \bar{a}) \quad \leftarrow \quad & \mathbf{B}, \texttt{all}(z), \texttt{clock}(x_i, s, z, v), \\ & \texttt{time}(u), v \leq u, \\ & \texttt{chosen}_R(x_i, s, z, \bar{a}, u). \end{aligned}$$

Let φ be the original non-ground rule of *pure*(P) (with head predicate R_{snd}) on which this ground rule is based. Let φ_2 be the original asynchronous rule of P on which φ is based (without relation `clock` but with the choice operator). The ground rule above gives rise to a valuation V such that $V(pos(\varphi_2)|_{sch(P)}) \subseteq M|^{x_i,s}$. The existence of $\psi \in G_M(P)$ implies additionally that $V(neg(\varphi_2)) \cap M = \emptyset$ and thus $V(neg(\varphi_2)) \cap M|^{x_i,s} = \emptyset$ because $M|^{x_i,s} \subseteq M$. Let φ_3 denote the simplification of φ_2, which is in *async*(P). Using Lemma F.19 we know that $D^{\uparrow s} = M|^{x_i,s}$ and thus $V(pos(\varphi_3)) \subseteq D$ and $V(neg(\varphi_3)) \cap D = \emptyset$. Therefore the rule φ_3 produces $R(z, \bar{a}) \in async(P)(D)$. Therefore $\langle k, \boldsymbol{f} \rangle \in tag(i, \delta)$.

For the converse containment, let $\langle k, \boldsymbol{f} \rangle \in b^{\rho}$. We show that $\langle k, \boldsymbol{f} \rangle \in b^{\rho_{i+1}}$. Again, fact $\boldsymbol{f}$ is of the form $R(z, \bar{a})$.

- Suppose that $\langle k, \boldsymbol{f} \rangle \in (b^{\rho_i} \setminus m_i)$. To rephrase, this means $\langle k, \boldsymbol{f} \rangle \in b^{\rho_i}$ and $\langle k, \boldsymbol{f} \rangle \notin m_i$. By definition of b^{ρ_i}, the former implies that there is a fact $R_{\text{snd}}(y, t, z, u, \bar{a}) \in M$ such that $k = ord(y, t) < i \leq ord(z, u)$. The latter implies that $ord(z, u) \neq i$. Taking these two things together, we obtain $k = ord(y, t) < i < ord(z, u)$ and thus $ord(y, t) < i + 1 \leq ord(z, u)$. By definition of $b^{\rho_{i+1}}$ we then have $\langle k, \boldsymbol{f} \rangle \in b^{\rho_{i+1}}$.
- Suppose that $\langle k, \boldsymbol{f} \rangle \in tag(i, \delta)$. Then $k = i$. Let φ and V be a simplified asynchronous rule of *async*(P) and valuation that produced $\boldsymbol{f}$. Let φ_2 denote the original asynchronous rule in P on which φ is based. Let V_2 be V extended to assign s to the body timestamp-variable. Note V_2 is only partial because we don't say what value should be assigned to the time variable in the head. We have $V(pos(\varphi)) \subseteq D$ and $V(neg(\varphi)) \cap D = \emptyset$. Using Lemma F.19 we know that $D^{\uparrow s} = M|^{x_i,s}$. This implies that $V_2(pos(\varphi_2)|_{sch(P)}) \subseteq M|^{x_i,s} \subseteq M$ and $V_2(neg(\varphi_2)) \cap M|^{x_i,s} = \emptyset$. Because the facts in $V_2(neg(\varphi_2))$ are over $sch(P)^{\text{LT}}$ and have location specifier x_i and timestamp s, we have $V_2(neg(\varphi_2)) \cap M = \emptyset$. Let φ_3 be the R_{snd}-rule in *pure*(P) based on φ_2, where φ_3 has the form (4.8). There exists a ground rule ψ based on φ_3 and V_2:

$$\begin{aligned} R_{\text{snd}}(x_i, s, z, u, \bar{a}) \quad \leftarrow \quad & V_2(pos(\varphi_2)|_{sch(P)}), V_2(neq(\varphi_2)), \texttt{all}(z), \\ & \texttt{clock}(x_i, s, z, v), \texttt{time}(u), v \leq u, \\ & \texttt{chosen}_R(x_i, s, z, \bar{a}, u). \end{aligned}$$

  where v is a value that can be chosen such that $\texttt{clock}(x_i, s, z, v) \in M$ (Corollary F.8) and there exists some value $u \geq v$ such that $\texttt{chosen}_R(x_i, s, z, \bar{a}, u) \in M$ (Lemma F.10). The nonequalities $neq(\varphi_2)$ must be satisfied under V_2 because they are satisfied under V. We have $\psi \in G_M(P)$ because $V_2(neg(\varphi_2)) \cap M = \emptyset$. Thus we derive $R_{\text{snd}}(x_i, s, z, u, \bar{a}) \in M$ because M is a fixpoint. Using Lemma F.14, we have $v_M(x_i, s) < v_M(z, u)$. Therefore $ord(x_i, s) < ord(z, u)$ and thus $i < ord(z, u)$. We obtain $ord(x_i, s) < i + 1 \leq ord(z, u)$. Now, by definition of $b^{\rho_{i+1}}$ we have $\langle k, \boldsymbol{f} \rangle = \langle i, \boldsymbol{f} \rangle \in b^{\rho_{i+1}}$.

□

Lemma F.19. *Let $i \in \mathbb{N}$. Let $(\rho_i, x_i, m_i, i, \rho_{i+1})$ denote the tuple i of $\mathcal{R}$. Denote $S = s^{\rho_i}|^{x_i} \cup untag(m_i)$, $D = deduc(P)(S)$ and $s = local_C(i, x_i)$. We have $D^{\uparrow s} = M|^{x_i,s}$.*

Proof. By Lemma F.23 we have $ord(x_i, s) = i$.

For a stratum number k, denote $D_k = deduc_k(P)(S)$. We show by induction on the stratum numbers k that $(D_k)^{\uparrow s} = M_k|^{x,s}$. We will show later that $M_l|^{x,s} = M|^{x,s}$ with l the highest stratum number of *deduc*(P).

Base case: $k = 0$. . This stratum is empty by assumption (see Section A). Therefore by definition $D_k = S$ and $M_k|^{x_i,s} = M^{\blacktriangle}|^{x_i,s}$. Recall by definition that

$$\begin{aligned} M^{\blacktriangle} \quad = \quad & M|_{edb(P)^{\text{LT}}} \cup M^{\text{ind}} \\ & \cup \{R(y, t, \bar{a}) \mid \exists z, u : R_{\text{snd}}(z, u, y, t, \bar{a}) \in M\}. \end{aligned}$$

Since

$$M|_{edb(P)^{\text{LT}}} = \bigcup_{z \in \mathcal{N}} \bigcup_{t \in \mathbb{N}} (H|^z)^{\uparrow t},$$

we have

$$\begin{aligned} M^{\blacktriangle}|^{x_i,s} \quad = \quad & (H|^{x_i})^{\uparrow s} \cup M^{\text{ind}}|^{x_i,s} \\ & \cup \{R(x_i, s, \bar{a}) \mid \exists z, u : R_{\text{snd}}(z, u, x_i, s, \bar{a}) \in M\}. \end{aligned}$$

By definition, we have $m_i = pairs(\text{deliv}_i)$ with

$$\text{deliv}_i = \{R_{\text{snd}}(z, u, y, t, \bar{b}) \in M \mid ord(y, t) = i\}.$$

But in this expression it must be that $(y, t) = (x_i, s)$ because $ord(x_i, s) = i$ and ord is injective. Then by definition of *pairs*(deliv_i) and *untag*(m_i) we have

$$untag(m_i)^{\uparrow s} = \{R(x_i, s, \bar{a}) \mid \exists z, u : R_{\text{snd}}(z, u, x_i, s, \bar{a}) \in M\}.$$

Next, by definition, we have $\mathit{state}(x_i, s) = H|^{x_i} \cup (M^{\mathrm{ind}}|^{x_i,s})^{\downarrow}$. Also, from the definition of s^{ρ_i} it follows that $s^{\rho_i}|^{x_i} = \mathit{state}(x_i, s)$. Therefore, by rewriting the expression for $M^{\blacktriangle}|^{x_i,s}$ above, we find $M^{\blacktriangle}|^{x_i,s} = (s^{\rho_i}|^{x_i})^{\uparrow s} \cup \mathit{untag}(m_i)^{\uparrow s} = S^{\uparrow s}$.

Finally, by combining everything, we find $(D_k)^{\uparrow s} = S^{\uparrow s} = M^{\blacktriangle}|^{x_i,s} = M_k|^{x_i,s}$.

Induction hypothesis.

Assume that the property holds up to and including stratum $k-1$ (with $k-1 \geq 0$), thus $(D_{k-1})^{\uparrow s} = M_{k-1}|^{x,s}$.

Inductive step.

We show that the property holds for stratum k.

First we show that $(D_k)^{\uparrow s} \subseteq M_k|^{x_i,s}$. We can consider the fixpoint calculation of D_k for stratum k that is obtained by deriving one fact at the time. This gives us a sequence $D_k^0 \subseteq D_k^1 \subseteq D_k^2 \ldots$ of fact-sets with $D_k^0 = D_{k-1}$. We show by induction on $j = 0, 1, 2, \ldots$ that $(D_k^j)^{\uparrow s} \subseteq M_k|^{x_i,s}$. For the base case we have $D_k^0 = D_{k-1}$ and thus $(D_k^0)^{\uparrow s} = (D_{k-1})^{\uparrow s} = M_{k-1}|^{x_i,s} \subseteq M_k|^{x_i,s}$ by the outer induction hypothesis. For the induction hypothesis we assume that $(D_k^{j-1})^{\uparrow s} \subseteq M_k|^{x_i,s}$ with $j-1 \geq 0$. For the inductive step we show that $(D_k^j)^{\uparrow s} \subseteq M_k|^{x_i,s}$. Let $\boldsymbol{f} \in D_k^j \setminus D_k^{j-1}$. Let φ and V be a simplified deductive rule of $\mathit{deduc}_k(P)$ and valuation V that have derived $\boldsymbol{f}$. Let φ_2 be the original, unsimplified version of φ in P. Let V_2 be the valuation for φ_2, obtained by extending V to assign value s to the timestamp variable of the body, which is the same timestamp variable as in the head because φ_2 is deductive. Let ψ be the deductive ground rule based on φ_2 and V_2 (ψ has no negative body literals). We next show that $\psi \in G_M(P)$. Because V is satisfying for φ, we have $V(\mathit{pos}(\varphi)) \subseteq D_k^{j-1}$, $V(\mathit{neg}(\varphi)) \cap D_{k-1} = \emptyset$ and the nonequalities of φ are satisfied.[2] By applying the inner induction hypothesis, we find that $V_2(\mathit{pos}(\varphi_2)) \subseteq (D_k^{j-1})^{\uparrow s} \subseteq M_k|^{x_i,s} \subseteq M$. Also, $V_2(\mathit{neg}(\varphi_2)) \cap (D_{k-1})^{\uparrow s} = \emptyset$. By applying the outer induction hypothesis we have that $V_2(\mathit{neg}(\varphi_2)) \cap M_{k-1}|^{x_i,s} = \emptyset$. Suppose that there is some fact $\boldsymbol{g} \in V_2(\mathit{neg}(\varphi_2)) \cap M$. We show that this leads to a contradiction, so that $V_2(\mathit{neg}(\varphi_2)) \cap M = \emptyset$. Because φ_2 is deductive, note that $\boldsymbol{g}$ is over the schema $\mathit{sch}(P)^{\mathrm{LT}}$ and has location specifier x_i and timestamp s.

- Suppose that $\boldsymbol{g} \in M$ is derived by a deductive ground rule. Because negation in φ_2 can only be applied to lower strata, we must have $\boldsymbol{g} \in M_{k-1}|^{x_i,s}$, a contradiction.
- Suppose that $\boldsymbol{g} \in M$ is derived by an inductive ground rule. Then $\boldsymbol{g} \in M^{\mathrm{ind}}|^{x_i,s}$. By definition of $\mathit{state}(x_i, s)$ and s^{ρ_i}, we would then have $\boldsymbol{g}^{\downarrow} \in \mathit{state}(x_i, s) \subseteq s^{\rho_i}|^{x_i} \subseteq S \subseteq D_{k-1}$ and thus $\boldsymbol{g} \in (D_{k-1})^{\uparrow s}$, a contradiction.
- Suppose that $\boldsymbol{g} \in M$ is derived by a rule of the form

  $$\boldsymbol{g} \leftarrow R_{\mathrm{snd}}(z, u, x_i, s, \bar{a}).$$

  Then $\boldsymbol{g}^{\downarrow} \in \mathit{untag}(m_i) \subseteq S \subseteq D_{k-1}$, again a contradiction.

We thus find that $V_2(\mathit{neg}(\varphi_2)) \cap M = \emptyset$. Therefore the rule ψ exists in $G_M(P)$ and it derives $\mathit{head}(\psi) = \boldsymbol{f}^{\uparrow s} \in M_k|^{x_i,s}$.

Now we show that $M_k|^{x_i,s} \subseteq (D_k)^{\uparrow s}$. Because M is a fixpoint of $G_M(P)$, we consider the fixpoint calculation of M to be a sequence of fact sets $M^0 \subseteq M^1 \subseteq M^2 \subseteq \ldots$ where $M^0 = \mathit{decl}(H)$ and where we only derive one new fact at the time. We show by induction on $j \in \mathbb{N}$ that $M_k^j|^{x_i,s} \subseteq (D_k)^{\uparrow s}$. This results in $M_k|^{x_i,s} = M_k^{\infty}|^{x_i,s} \subseteq (D_k)^{\uparrow s}$.

For the base case, there are no deductively or inductively derived facts nor R_{snd}-message facts in $M^0 = \mathit{decl}(H)$. Therefore $M_k^0 = M^0|_{\mathit{edb}(P)^{\mathrm{LT}}} = \bigcup_{z \in \mathcal{N}} \bigcup_{t \in \mathbb{N}} (H|^z)^{\uparrow t}$. Thus $M_k^0|^{x_i,s} = (H|^{x_i})^{\uparrow s} \subseteq \mathit{state}(x_i, s)^{\uparrow s} \subseteq S^{\uparrow s} \subseteq (D_k)^{\uparrow s}$.

For the induction hypothesis we assume that $M_k^{j-1}|^{x_i,s} \subseteq (D_k)^{\uparrow s}$ with $j-1 \geq 0$. For the inductive step we show that $M_k^j|^{x_i,s} \subseteq (D_k)^{\uparrow s}$. Let $\boldsymbol{g} \in (M_k^j \setminus M_k^{j-1})|^{x_i,s}$. Let $\psi \in G_M(P)$ be the deductive ground rule that derived $\boldsymbol{g}$. If the stratum of ψ is $k-1$ or smaller, then $\boldsymbol{g} \in M_{k-1}|^{x_i,s}$ and then we can apply the outer induction hypothesis to know that $\boldsymbol{g} \in (D_{k-1})^{\uparrow s} \subseteq (D_k)^{\uparrow s}$. Now assume that ψ has stratum k. Since the body of ψ is true on M^{j-1} and ψ is deductive, we specifically have $\mathit{pos}(\psi) \subseteq M^{j-1}|^{x_i,s}$. Now for $\boldsymbol{f} \in \mathit{pos}(\psi)$ we show that $\boldsymbol{f} \in (D_k)^{\uparrow s}$. Note that the location specifier and timestamp in $\boldsymbol{f}$ are x_i and s respectively. Let h be the index such that $\boldsymbol{f} \in M^h \setminus M^{h-1}$, which implies $h < j$. There are three ways in which $\boldsymbol{f}$ could have been derived:

- Suppose that $\boldsymbol{f}$ was derived by a deductive ground rule ψ_2. Because the deductive rules of $\mathit{pure}(P)$ are syntactically stratified, $\mathit{stratum}(\psi_2) \leq \mathit{stratum}(\psi) = k$. Denote $l = \mathit{stratum}(\psi_2)$. We have $\mathit{pos}(\psi_2) \subseteq M^{h-1} \subseteq M^h$. Then $\boldsymbol{f} \in M_l^h$ by definition of M_l^h. Now since $M^h \subseteq M^{j-1}$ we have $\boldsymbol{f} \in M_l^{j-1}$. Also, since $l \leq k$ we have $\boldsymbol{f} \in M_k^{j-1}|^{x_i,s}$. By applying the inner induction hypothesis we now obtain $\boldsymbol{f} \in (D_k)^{\uparrow s}$.
- Suppose that $\boldsymbol{f} \in M$ is derived by an inductive ground rule. Then $\boldsymbol{f} \in M^{\mathrm{ind}}|^{x_i,s}$ and thus $\boldsymbol{f}^{\downarrow} \in H|^{x_i} \cup (M^{\mathrm{ind}}|^{x_i,s})^{\downarrow} = \mathit{state}(x_i, s)$. Then, by definition of s^{ρ_i} and S, we have $\boldsymbol{f}^{\downarrow} \in \mathit{state}(x_i, s) \subseteq s^{\rho_i}|^{x_i} \subseteq S \subseteq D_{k-1}$. Therefore $\boldsymbol{f} \in (D_{k-1})^{\uparrow s} \subseteq (D_k)^{\uparrow s}$.
- Suppose that $\boldsymbol{f} \in M$ is derived by a ground rule of the form

  $$\boldsymbol{f} \leftarrow R_{\mathrm{snd}}(z, u, x_i, s, \bar{a}).$$

  Then $\boldsymbol{f}^{\downarrow} \in \mathit{untag}(m_i) \subseteq S \subseteq D_{k-1}$ and thus $\boldsymbol{f} \in (D_{k-1})^{\uparrow s} \subseteq (D_k)^{\uparrow s}$.

Thus overall, $\mathit{pos}(\psi) \subseteq (D_k)^{\uparrow s}$. Next, because $\psi \in G_M(P)$ exists, it is possible to choose an original deductive rule φ of P and a valuation V for φ such that ψ is the ground rule based on φ and V (without negative body literals) and such that

[2] Recall that in stratified programs negation is applied only to lower strata.

$V(neg(\varphi)) \cap M = \emptyset$. Since $M_{k-1}|^{x_i,s} \subseteq M$ we also have $V(neg(\varphi)) \cap M_{k-1}|^{x_i,s} = \emptyset$. By now applying the outer induction hypothesis we obtain $V(neg(\varphi)) \cap (D_{k-1})^{\uparrow s} = \emptyset$. Let φ' be the simplified rule in $deduc_k(P)$ that is based on φ and let V' be the accompanying simplification of V. We now have $V'(pos(\varphi')) \subseteq D_k$ and $V'(neg(\varphi')) \cap D_{k-1} = \emptyset$. We thus obtain that during the computation of stratum k of D, the rule φ' under valuation V' derives $head(V'(\varphi')) = \boldsymbol{g}^{\downarrow} \in D_k$. Thus $\boldsymbol{g} \in (D_k)^{\uparrow s}$.

Last step.

It remains to show that $M_l|^{x_i,s} = M|^{x_i,s}$ where l is the highest stratum number. Surely $M_l|^{x_i,s} \subseteq M|^{x_i,s}$. Now, assuming the existence of a fact $\boldsymbol{g} \in (M \setminus M_l)|^{x_i,s}$ can be shown to lead to a contradiction, in a similar manner as done in the proof above:

- If $\boldsymbol{g} \in M$ were derived by a deductive ground rule, we immediately get $\boldsymbol{g} \in M_l$ because the stratum number of this rule must be less than or equal to l.
- Suppose that $\boldsymbol{g} \in M$ is derived by an inductive ground rule. Then $\boldsymbol{g} \in M^{\text{ind}}|^{x_i,s}$. By definition of $state(x_i, s)$ and s^{ρ_i}, we would then have $\boldsymbol{g}^{\downarrow} \in s^{\rho_i}|^{x_i} \subseteq S \subseteq D_l$ and thus $\boldsymbol{g} \in (D_l)^{\uparrow s} = M_l|^{x_i,s}$, a contradiction.
- Suppose that $\boldsymbol{g} \in M$ is derived by a rule of the form

  $$\boldsymbol{g} \leftarrow R_{\text{snd}}(z, u, x_i, s, \bar{a}).$$

  Then $\boldsymbol{g}^{\downarrow} \in untag(m_i) \subseteq S \subseteq D_l$, and like above, this is a contradiction.

We obtain that $D^{\uparrow s} = M|^{x_i,s}$. ☐

Lemma F.20. *Let i be a global transition index of $\mathcal{R}$. We have $b^{\rho_{i+1}} = (b^{\rho_i} \setminus m_i) \cup tag(i, \delta_i)$.*

Proof. This follows from Lemma F.18. ☐

Lemma F.21. *M is the trace of the run $\mathcal{R}$.*

Proof. Let N denote the trace of $\mathcal{R}$, as defined in Section 5. Using Proposition E.9 we know that N is a stable model. Using Lemma F.22 we then know that the stable model N is included in the stable model M ($N \subseteq M$). We now show that $M \subseteq N$ as well.

We can imagine that M is obtained by executing the ground rules of $G_M(P)$ one by one on input $decl(H)$. This gives us a sequence of fact sets $M_0 \subseteq M_1 \subseteq M_2 \subseteq \ldots$ with $M_0 = decl(H)$ and $M_\infty = \bigcup_i M_i = M$. We show by induction on $i = 0, 1, 2, \ldots$ that $M_i \subseteq N$. For the base case we have $M_0 = decl(H) \subseteq N$ by Lemma D.5. For the induction hypothesis, assume that $M_{i-1} \subseteq N$ with $i - 1 \geq 0$. For the inductive step, let $\{\boldsymbol{f}\} = M_i \setminus M_{i-1}$. We show that $\boldsymbol{f} \in N$. Let $\psi \in G_M(P)$ be the ground rule that derived $\boldsymbol{f}$ in M_i. We have $pos(\psi) \subseteq M_{i-1}$ and the nonequalities of ψ are satified. Let φ and V be an original rule of $pure(P)$ and valuation that give rise to the ground rule ψ. Because ψ is in $G_M(P)$ we can choose φ and V so that $V(neg(\varphi)) \cap M = \emptyset$. Because $N \subseteq M$ we have $V(neg(\varphi)) \cap N = \emptyset$. Therefore ψ is in $G_N(P)$ too. Using the induction hypothesis we obtain that $pos(\psi) \subseteq M_{i-1} \subseteq N$ and because the nonequalities are satisfied, we obtain that ψ derives $\boldsymbol{f} \in N$ because N is a stable model. ☐

Lemma F.22. *The trace of $\mathcal{R}$ is included in M.*

Proof. Let $N = \bigcup_i N_i$ denote the trace of $\mathcal{R}$, as defined in Section 5 (where we have used notation M_i for N_i). We show by induction on $i \in \mathbb{N} \cup \{-1\}$ that $N_i \subseteq M$. For the base case ($i = -1$), we have $N_{-1} \subseteq M$ by Lemmas F.1 and F.2 and because $decl(H) \subseteq M$. For the induction hypothesis we assume that $N_{i-1} \subseteq M$ with $i - 1 \geq -1$. For the inductive step we show that $N_i \subseteq M$.

Let $(x_i, s) \in L$ be such that $ord(x_i, s) = i$. By Lemma F.23, we have $local_C(i, x_i) = s$, or equivalently $local_{\mathcal{R}}(i) = s$, using the notation from Section 5.1. Therefore by Lemma F.24, for $j \in prev_{\mathcal{R}}(i)$ we have $j = ord(x_i, s - 1)$.

Let α be the unique arrival function based on $\mathcal{R}$. For each of the sets that comprise $N_i \setminus N_{i-1}$ we show containment in M.

State (5.1).

For the set (5.1) of N_i, we show that $D_i^{\uparrow s} \subseteq M$. Using Lemma F.19 we have $D_i^{\uparrow s} = M|^{x_i,s} \subseteq M$.

Clock (5.2).

For the set (5.2) of N_i, we show $\{\mathtt{rcvClock}(x_i, s, x_i, s+1)\} \subseteq M$. This set is in M by Lemma F.2.

Clock (5.3).

For the set (5.3) of N_i, we show that M contains

$$\{\mathtt{rcvClock}(x_i, s, y, t) \mid y \in \mathcal{N},\ x_i \neq y,\ t \in \pi_{\mathcal{R}}(i, y)\}.$$

Let $y \in \mathcal{N}$ with $x_i \neq y$. By definition, $\pi_{\mathcal{R}}(i, y) = \{v_{\mathcal{R}}(j)[y] \mid j \in prev_{\mathcal{R}}(i)\}$ (see Section (5.1)). Suppose that $prev_{\mathcal{R}}(i) \neq \emptyset$ and let $j \in prev_{\mathcal{R}}(i)$. This implies $s > 0$. From above, we have $j = ord(x_i, s-1)$. By Lemma F.26 we have $v_M(x_i, s-1) = v_{\mathcal{R}}(j)$.

By definition of $v_M(x_i, s-1)$, we have $\texttt{clock}(x_i, s-1, y, t) \in M$ with $t = v_M(x_i, s-1)[y]$. Now if $x_i \neq y$, the following ground rule, based on the rule (4.5), is in $G_M(P)$ and it derives $\texttt{rcvClock}(x_i, s, y, t) \in M$ because M is a fixpoint:

$$\begin{aligned}\texttt{rcvClock}(x_i, s, y, t) \quad \leftarrow \quad & \texttt{clock}(x_i, s-1, y, t), x_i \neq y, \\ & \texttt{tsucc}(s-1, s).\end{aligned}$$

Clock (5.4).

Let $\texttt{rcvClock}(x_i, s, y, t) \in N_i$ and $\texttt{rcvClock}(x_i, s, y, t') \in N_i$ with $t < t'$. By definition of the set (5.4) of N_i, we have $\texttt{isBehind}(x_i, s, y, t) \in N_i$. Now we show that $\texttt{isBehind}(x_i, s, y, t) \in M$.

Consider the fact $\texttt{rcvClock}(x_i, s, y, t) \in N_i$. This fact is in N_{i-1} or $N_i \setminus N_{i-1}$.

- If it is in N_{i-1} then we can apply the induction hypothesis to know that $\texttt{rcvClock}(x_i, s, y, t) \in M$.
- If it is in $N_i \setminus N_{i-1}$, then it can not be in the set (5.9) of N_i because if there is a fact in this set with first component x_i, then the second component must be at least $s+1$ by Lemma D.4. So, if $\texttt{rcvClock}(x_i, s, y, t)$ is in $N_i \setminus N_{i-1}$, then it is in the set (5.2) or the set (5.3). And for both of these sets we have shown above that then $\texttt{rcvClock}(x_i, s, y, t) \in M$.

Similarly we can show that $\texttt{rcvClock}(x_i, s, y, t') \in M$. Now, the following ground rule is in $G_M(P)$, based on the rule (4.6):

$$\begin{aligned}\texttt{isBehind}(x, s, y, t) \quad \leftarrow \quad & \texttt{rcvClock}(x, s, y, t), \\ & \texttt{rcvClock}(x, s, y, t'), t < t'.\end{aligned}$$

Now since M is a fixpoint, we have $\texttt{isBehind}(x, s, y, t) \in M$.

Clock (5.5).

For the set (5.5) of N_i we show that M contains:

$$\{\texttt{clock}(x_i, s, y, t) \mid y \in \mathcal{N},\ t = v_{\mathcal{R}}(i)[y]\}.$$

Let $y \in \mathcal{N}$. By Lemma F.26 we have $v_{\mathcal{R}}(i) = v_M(x_i, s)$, which implies $v_{\mathcal{R}}(i)[y] = v_M(x_i, s)[y]$. Now by definition of $v_M(x_i, s)[y] = t$ this implies $\texttt{clock}(x_i, s, y, t) \in M$.

Sending (5.6).

For the set (5.6) of N_i we show that M contains

$$\{R_{\text{snd}}(x_i, s, y, t, \bar{a}) \mid \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t = \mathit{local}_{\mathcal{R}}(\alpha(i, \boldsymbol{f}))\}.$$

Let $\boldsymbol{f} = R(y, \bar{a}) \in \delta_i$. Denote $j = \alpha(i, \boldsymbol{f})$ and $t = \mathit{local}_{\mathcal{R}}(j)$. By definition of j, we have $\langle i, \boldsymbol{f} \rangle \in m_j$. Then by definition of m_j, there is a fact $R_{\text{snd}}(z, u, y, v, \bar{a}) \in M$ with $\mathit{ord}(z, u) = i$ and $\mathit{ord}(y, v) = j$. We now show that $R_{\text{snd}}(z, u, y, v, \bar{a}) = R_{\text{snd}}(x_i, s, y, t, \bar{a})$. First, because ord is injective we have $(z, u) = (x_i, s)$. Because $\mathit{ord}(y, v) = j$ and node y is the recipient during global transition j, by Lemma F.23 we obtain $v = \mathit{local}_C(j, y) = \mathit{local}_{\mathcal{R}}(j, y) = \mathit{local}_{\mathcal{R}}(j) = t$. Thus $R_{\text{snd}}(x_i, s, y, t, \bar{a}) = R_{\text{snd}}(z, u, y, v, \bar{a}) \in M$.

Sending (5.7).

For the set (5.7) of N_i we show that M contains

$$\{\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \mid \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t = \mathit{local}_{\mathcal{R}}(\alpha(i, \boldsymbol{f}))\}.$$

Let $\boldsymbol{f} = R(y, \bar{a}) \in \delta_i$. Denote $j = \alpha(i, \boldsymbol{f})$ and $t = \mathit{local}_{\mathcal{R}}(j)$. Because $R_{\text{snd}}(x_i, s, y, t, \bar{a}) \in M$ from above, there must be a ground rule $G_M(P)$ of the form (4.8) that derives this fact:

$$\begin{aligned}\psi: \ R_{\text{snd}}(x_i, s, y, t, \bar{a}) \quad \leftarrow \quad & \mathbf{B}, \texttt{all}(y), \texttt{clock}(x_i, s, y, u), \\ & \texttt{time}(t), u \leq t, \\ & \texttt{chosen}_R(x_i, s, y, \bar{a}, t).\end{aligned}$$

Therefore $\texttt{chosen}_R(x_i, s, y, \bar{a}, t) \in M$.

Sending (5.8).

For the set (5.8) of N_i we show that M contains

$$\begin{aligned}\{\texttt{other}_R(x_i, s, y, \bar{a}, t') \quad \mid \quad & \boldsymbol{f} = R(y, \bar{a}) \in \delta_i,\ t' \in \mathbb{N}, \\ & v_{\mathcal{R}}(i)[y] \leq t',\ t' \neq \mathit{local}_{\mathcal{R}}(\alpha(i, \boldsymbol{f}))\}.\end{aligned}$$

Let $\boldsymbol{f} = R(y, \bar{a}) \in \delta_i$. Denote $j = \alpha(i, \boldsymbol{f})$ and $t = \mathit{local}_{\mathcal{R}}(j)$. We have $v_{\mathcal{R}}(i) = v_M(x_i, s)$ by Lemma F.26. Denote $v = v_{\mathcal{R}}(i)[y]$. Thus $v = v_M(x_i, s)[y]$, and by definition of $v_M(x_i, s)$, we have $\texttt{clock}(x_i, s, y, v) \in M$. Take a value $t' \in \mathbb{N}$ with $v \leq t'$ and $t' \neq t$. We show that $\texttt{other}_R(x_i, s, y, \bar{a}, t') \in M$.

Based on the previously mentioned ground rule ψ with head-predicate R_{snd} above, the following ground rule exists in $G_M(P)$, based on the form (4.10):

$$\begin{aligned}\psi_2: \ \mathtt{other}_R(x_i, s, y, \bar{a}, t') \quad \leftarrow \quad & \mathbf{B}, \mathtt{all}(y), \mathtt{clock}(x_i, s, y, u), \\ & \mathtt{time}(t'), u \leq t', \\ & \mathtt{chosen}_R(x_i, s, y, \bar{a}, t), t \neq t'.\end{aligned}$$

For the set (5.7) above we have already shown $\mathtt{chosen}_R(x_i, s, y, \bar{a}, t) \in M$. Because the body of the ground rule ψ is true on M, we have $\mathtt{clock}(x_i, s, y, u) \in M$ and thus by Corollary F.8 we know $u = v$. Thus the restriction $u \leq t'$ in ψ_2 holds because $v \leq t'$ by assumption. Therefore the ground rule ψ_2 derives $\mathtt{other}_R(x_i, s, y, \bar{a}, t') \in M$ because M is a fixpoint.

Sending (5.9).

For the set (5.9) of N_i we show that M contains

$$\{\mathtt{rcvClock}(y, t, z, u) \ \ | \ \ \boldsymbol{f} = R(y, \bar{a}) \in \delta_i, \ t = \mathit{local}_{\mathcal{R}}(\alpha(i, \boldsymbol{f})), \ z \in \mathcal{N}, \ u = v_{\mathcal{R}}(i)[z]\}.$$

Let $\boldsymbol{f} = R(y, \bar{a}) \in \delta_i$. Denote $j = \alpha(i, \boldsymbol{f})$ and $t = \mathit{local}_{\mathcal{R}}(j)$. By Lemma F.26 we have $v_{\mathcal{R}}(i) = v_M(x_i, s)$. Let $z \in \mathcal{N}$. Denote $u = v_{\mathcal{R}}(i)[z]$. Thus $u = v_M(x_i, s)[z]$. By definition of $v_M(x_i, s)$, we have $\mathtt{clock}(x_i, s, z, u) \in M$.

We have shown above for the set (5.6) that $R_{\text{snd}}(x_i, s, y, t, \bar{a}) \in M$. Now, based on the rule (4.12), the following ground rule in $G_M(P)$ derives $\mathtt{rcvClock}(y, t, z, u) \in M$ because M is a fixpoint:

$$\begin{aligned}\mathtt{rcvClock}(y, t, z, u) \quad \leftarrow \quad & R_{\text{snd}}(x_i, s, y, t, \bar{a}), \\ & \mathtt{clock}(x_i, s, z, u).\end{aligned}$$

□

Lemma F.23. *Let $(x, s) \in L$. Denote $i = \mathit{ord}(x, s)$. We have $s = \mathit{local}_C(i, x)$.*

Proof. First, consider the following sets

$$A = \{(x, s') \in L \mid \mathit{ord}(x, s') < i\};$$

$$B = \{(x, s') \in L \mid s' < s\}.$$

We show that $A = B$, by showing both inclusions separately:

- We show that $A \subseteq B$. Let $(x, s') \in A$. Then $\mathit{ord}(x, s') < i$ and thus $\mathit{ord}(x, s') < \mathit{ord}(x, s)$. We show that if we would suppose that $s \leq s'$, we obtain a contradiction. So, suppose $s \leq s'$. Then $v_M(x, s) \preceq v_M(x, s')$ by Lemma F.13. Then $(x, s) \preceq (x, s')$. Since $\leq$ on L respects $\preceq$, we have $(x, s) \leq (x, s')$ and thus $\mathit{ord}(x, s) \leq \mathit{ord}(x, s')$, which is a contradiction. Therefore it must be that $s' < s$ and thus $(x, s') \in B$.
- We show that $B \subseteq A$. Let $(x, s') \in B$. We have $s' < s$ and thus by Lemma F.13 we have $v_M(x, s') \prec v_M(x, s)$. Then, $v_M(x, s') \preceq v_M(x, s)$ and $v_M(x, s') \neq v_M(x, s)$ imply respectively $(x, s') \preceq (x, s)$ and $(x, s') \neq (x, s)$. By definition of *ord* we then have respectively $\mathit{ord}(x, s') \leq \mathit{ord}(x, s)$ and $\mathit{ord}(x, s') \neq \mathit{ord}(x, s)$ (because *ord* is injective). Therefore $\mathit{ord}(x, s') < \mathit{ord}(x, s) = i$ and thus $(x, s') \in A$.

For every $n \in \mathbb{N}$ we have $n = |\{m \in \mathbb{N} \mid m < n\}|$. Now we can combine everything:

$$\begin{aligned}\mathit{local}_C(i, x) &= |A| \qquad \text{(by definition)} \\ &= |B| \\ &= |\{s' \in \mathbb{N} \mid s' < s\}| \qquad \text{(by def. of } L\text{)} \\ &= s.\end{aligned}$$

□

Lemma F.24. *Let i be a global transition index of $\mathcal{R}$. Let $(x_i, s) \in L$ be such that $\mathit{ord}(x_i, s) = i$. For $j \in \mathit{prev}_{\mathcal{R}}(i)$ we have $j = \mathit{ord}(x_i, s - 1)$.*

Proof. Let $j \in \mathit{prev}_{\mathcal{R}}(i)$. Let $(x_i, t) \in L$ be such that $\mathit{ord}(x_i, t) = j$. We show that $t = s - 1$.

We first show that $t < s$. Because by definition of $\mathit{prev}_{\mathcal{R}}(i)$ we have $j < i$, we obtain $\mathit{ord}(x_i, t) < \mathit{ord}(x_i, s)$. As a proof by contradiction, suppose that $t \geq s$. Then $v_M(x_i, s) \preceq v_M(x_i, t)$ by Lemma F.13 and thus $(x_i, s) \preceq (x_i, t)$. Then by definition of *ord* we have $\mathit{ord}(x_i, s) \leq \mathit{ord}(x_i, t)$, which is a contradiction. Hence $t < s$.

We will now show that $t = s - 1$. Again, as a proof by contradiction, suppose that $t < s - 1$. Then there is a value $t' \in \mathbb{N}$ such that $t < t' < s$. By Lemma F.13 we have $v_M(x_i, t) \prec v_M(x_i, t') \prec v_M(x_i, s)$ and thus $(x_i, t) \prec (x_i, t') \prec (x_i, s)$. Also, then by definition of *ord* we have $\mathit{ord}(x_i, t) < \mathit{ord}(x_i, t') < \mathit{ord}(x_i, s)$. But then there would be a global transition h with $x_h = x_i$ and $j < h$ and thus $j \notin \mathit{prev}_{\mathcal{R}}(i)$ by definition of $\mathit{prev}_{\mathcal{R}}(i)$. We have arrived at the contradiction. We conclude $t = s - 1$. □

Lemma F.25. *Let i be a global transition index of $\mathcal{R}$. Let $(x_i, s) \in L$ be such that $ord(x_i, s) = i$. We have $s > 0$ iff $prev_{\mathcal{R}}(i) \neq \emptyset$.*

Proof. First we show that if $s > 0$ then $prev_{\mathcal{R}}(i) \neq \emptyset$. We have $(x_i, s-1) \in L$ and $v_M(x_i, s-1) \prec v_M(x_i, s)$ by Lemma F.13. Then $(x_i, s-1) \prec (x_i, s)$ and by definition of *ord* we have $ord(x_i, s-1) < ord(x_i, s)$. So there is at least one pair $(x_i, t) \in L$ ordered before (x_i, s) in C. This implies that there is a global transition $j < i$ with $x_j = x_i$. Therefore $prev_{\mathcal{R}}(i) \neq \emptyset$.

Now suppose $prev_{\mathcal{R}}(i) \neq \emptyset$. Let $j \in prev_{\mathcal{R}}(i)$. Let $(x_i, t) \in L$ be such that $ord(x_i, t) = j$. We have $j < i$ by definition of $prev_{\mathcal{R}}(i)$ and thus $ord(x_i, t) < ord(x_i, s)$. If $s \leq t$ then $v_M(x_i, s) \preceq v_M(x_i, t)$ by Lemma (F.13) and thus $(x_i, s) \preceq (x_i, t)$, $(x_i, s) \leq (x_i, t)$ and $ord(x_i, s) \leq ord(x_i, t)$, which is false. Thus $t < s$ and since $t \in \mathbb{N}$ by definition of L, this implies $s > 0$. ☐

Lemma F.26. *Let i be a global transition index of $\mathcal{R}$. Let $(x_i, s_i) \in L$ be such that $ord(x_i, s_i) = i$. We have $v_M(x_i, s_i) = v_{\mathcal{R}}(i)$ where $v_{\mathcal{R}}(i)$ is as defined in Section 5.1.*

Proof. Proof by induction on $i \in \mathbb{N}$.

- Base case: $i = 0$. We defined $v_{\mathcal{R}}(0)[x_0] = 1$ and $v_{\mathcal{R}}(0)[y] = 0$ for $y \neq x_0$. By Lemma F.23 we have $s_0 = local_C(0, x_0) = 0$. By Lemma F.12 we have $v_M(x_0, 0)[x_0] = 1$.
  Now consider $y \in \mathcal{N}$ with $y \neq x_0$. We must show that $v_M(x_0, 0)[y] = 0$. Denote $u = v_M(x_0, 0)[y]$. As a proof by contradiction, suppose that $u > 0$. We have $\texttt{clock}(x_0, 0, y, u) \in M$ by definition of $v_M(x_0, 0)$ and thus also $\texttt{rcvClock}(x_0, 0, y, u) \in M$ by looking at the rule (4.7). Because $s_0 = 0$, the only ground rule that could have derived this $\texttt{rcvClock}$-fact is of the form (4.12):

  $$\texttt{rcvClock}(x_0, 0, y, u) \leftarrow R_{\text{snd}}(z, v, x_0, 0, \bar{a}), \texttt{clock}(z, v, y, u).$$

  But then by Lemma F.14 we would have $v_M(z, u) \prec v_M(x_0, 0)$ and thus $(z, u) \prec (x_0, 0)$ and $ord(z, u) < ord(x_0, 0) = 0$. But $ord(z, u) < 0$ is impossible by definition of *ord*. Thus $u = 0$.
- Inductive step: consider transition i. We defined $v_{\mathcal{R}}(i)[x_i] = local_{\mathcal{R}}(i) + 1$. By construction of $\mathcal{R}$ out of C we have $local_{\mathcal{R}}(i) = local_C(i, x_i)$ and Lemma F.23 we have $local_C(i, x_i) = s_i$. Thus $local_{\mathcal{R}}(i) = s_i$. By Lemma F.12 we have $v_M(x_i, s_i)[x_i] = s_i + 1 = local_{\mathcal{R}}(i) + 1 = v_{\mathcal{R}}(i)[x_i]$.
  Next, let $y \in \mathcal{N}$ with $y \neq x_i$. Let α denote the arrival function for $\mathcal{R}$. We defined $v_{\mathcal{R}}(i)[y] = \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$ with (see Section (5.1)):

  $$\mu_{\mathcal{R}}(i, y) = \{v_{\mathcal{R}}(k)[y] \mid \exists \boldsymbol{f} : (k, \boldsymbol{f}) \in sent(\mathcal{R}), \alpha(k, \boldsymbol{f}) = i\},$$

  $$\pi_{\mathcal{R}}(i, y) = \{v_{\mathcal{R}}(j)[y] \mid j \in prev_{\mathcal{R}}(i)\}.$$

  We now proceed in several steps.

  1. We show that $v_M(x_i, s_i)[y] \geq \max(\{0\} \cup \pi_{\mathcal{R}}(i, y))$. If $s_i = 0$ then $prev_{\mathcal{R}}(i) = \emptyset$ by Lemma (F.25), and thus $\pi_{\mathcal{R}}(i, y) = \emptyset$ and the inequality trivially holds. For the case $s_i > 0$, we show that $v_M(x_i, s_i)[y] \geq \max(\pi_{\mathcal{R}}(i, y))$. Because $s_i > 0$ we have $prev_{\mathcal{R}}(i) \neq \emptyset$ by Lemma F.25. Let $j \in prev_{\mathcal{R}}(i)$. By Lemma F.24 we have $j = ord(x_i, s_i - 1)$. By using the induction hypothesis, we have $v_{\mathcal{R}}(j)[y] = v_M(x_i, s_i - 1)[y]$. Denote $u = v_M(x_i, s_i - 1)[y]$. By definition of $v_M(x_i, s_i - 1)$ we have $\texttt{clock}(x_i, s_i - 1, y, u) \in M$. The following ground rule derives $\texttt{rcvClock}(x_i, s_i, y, u) \in M$:

     $$\begin{aligned} \texttt{rcvClock}(x_i, s_i, y, u) \quad \leftarrow \quad & \texttt{clock}(x_i, s_i - 1, y, u), x_i \neq y, \\ & \texttt{tsucc}(s_i - 1, s_i). \end{aligned}$$

     Now by Lemma F.11, $\texttt{rcvClock}(x_i, s_i, y, u) \in M$ implies $v_M(x_i, s_i)[y] \geq u$. In summary, thus $v_M(x_i, s_i)[y] \geq \max(\{0\} \cup \pi_{\mathcal{R}}(i, y))$.
  2. Suppose $\mu_{\mathcal{R}}(i, y) \neq \emptyset$. We show that $v_M(x_i, s_i)[y] \geq \max(\mu_{\mathcal{R}}(i, y))$. Let $(k, \boldsymbol{f}) \in sent(\mathcal{R})$ be such that $\alpha(k, \boldsymbol{f}) = i$. Denote $u = v_{\mathcal{R}}(k)[y]$. We show that $v_M(x_i, s_i)[y] \geq u$. By definition of α we have $\langle k, \boldsymbol{f} \rangle \in m_i$. Because $i = ord(x_i, s_i)$ the fact $\boldsymbol{f}$ is of the form $R(x_i, \bar{a})$ (thus with location specifier x_i). By definition, $m_i = pairs(\text{deliv}_i)$ and thus there is a fact $R_{\text{snd}}(z, t, x_i, s_i, \bar{a}) \in \text{deliv}_i \subseteq M$ with $ord(z, t) = k$. By Lemma F.14 we have $v_M(z, t) \prec v_M(x_i, s_i)$ and thus $k = ord(z, t) < ord(x_i, s_i) = i$. We have the following ground rule in $G_M(P)$, based on rule (4.12):

     $$\texttt{rcvClock}(x_i, s_i, y, v) \leftarrow R_{\text{snd}}(z, t, x_i, s_i, \bar{a}), \texttt{clock}(z, t, y, v).$$

     By applying the induction hypothesis to k we have $v_{\mathcal{R}}(k)[y] = v_M(z, t)[y] = u$ and thus $v = u$. By definition of $v_M(z, t)[y]$ we have $\texttt{clock}(z, t, y, u) \in M$. The body of the previous ground rule is true on M. Now since M is a fixpoint we have $\texttt{rcvClock}(x_i, s_i, y, u) \in M$. By Lemma F.11 this implies $v_M(x_i, s_i)[y] \geq u$. Thus overall $v_M(x_i, s_i)[y] \geq \max(\mu_{\mathcal{R}}(i, y))$.
  3. An intermediate conclusion is now that $v_M(x_i, s_i)[y] \geq \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$.
  4. We now show that $v_M(x_i, s_i)[y] \leq \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$. Denote $u = v_M(x_i, s_i)[y]$. By definition of $v_M(x_i, s_i)$ we have $\texttt{clock}(x_i, s_i, y, u) \in M$ and thus $\texttt{rcvClock}(x_i, s_i, y, u) \in M$ by looking at rule (4.7). If $u = 0$ then clearly $u \leq \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$.
     Now suppose $u > 0$. We will specifically show that $v_M(x_i, s_i)[y] \leq \max(\mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$. We will now consider the possible ways in which the fact $\texttt{rcvClock}(x_i, s_i, y, u) \in M$ could have been generated. We can already eleminate the $\texttt{rcvClock}$-rules in $G_M(P)$ that force $u = 0$ or force $y = x_i$. Only the following forms of $\texttt{rcvClock}$-rules remain:

(a) The fact $\texttt{rcvClock}(x_i, s_i, y, u) \in M$ could have been generated by a ground rule of the following form, based on rule (4.5):

$$\begin{aligned}\texttt{rcvClock}(x_i, s_i, y, u) \quad \leftarrow \quad & \texttt{clock}(x_i, s_i - 1, y, u), x_i \neq y, \\ & \texttt{tsucc}(s_i - 1, s_i).\end{aligned}$$

This implies $s_i > 0$. Since the body of this ground rule is true, we have $\texttt{clock}(x_i, s_i - 1, y, u) \in M$. By definition of $v_M(x_i, s_i - 1)$ this implies $v_M(x_i, s_i - 1)[y] = u$. Since $v_M(x_i, s_i - 1) \prec v_M(x_i, s_i)$ by Lemma F.13, we have $ord(x_i, s_i - 1) < ord(x_i, s_i) = i$. Denote $j = ord(x_i, s_i - 1)$. By Lemmas F.25 and F.24 we have $j \in prev_{\mathcal{R}}(i)$. By applying the induction hypothesis to j we have $v_{\mathcal{R}}(j)[y] = v_M(x_i, s_i - 1)[y] = u$. Thus $u \in \pi_{\mathcal{R}}(i, y)$, so certainly $u \leq \max(\mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$.

(b) The other option is that the fact $\texttt{rcvClock}(x_i, s_i, y, u) \in M$ was generated by a ground rule of the following form, based on the rule (4.12):

$$\texttt{rcvClock}(x_i, s_i, y, u) \leftarrow R_{\text{snd}}(z, t, x_i, s_i, \bar{a}), \texttt{clock}(z, t, y, u).$$

First, since the body of this ground rule is true on M, we have $\texttt{clock}(z, t, y, u) \in M$ and this implies $v_M(z, t)[y] = u$, by definition of $v_M(z, t)$. By Lemma F.14 we have $v_M(z, t) \prec v_M(x_i, s_i)$ and therefore $ord(z, t) < ord(x_i, s_i) = i$. Denote $k = ord(z, t)$ and $\boldsymbol{f} = R(x_i, \bar{a})$. Because $k < ord(x_i, s_i)$ we have $k + 1 \leq ord(x_i, s_i)$. Thus by definition of buf_k and buf_{k+1}, we have $R_{\text{snd}}(z, t, x_i, s_i, \bar{a}) \in \text{buf}_{k+1} \setminus \text{buf}_k$. Because there are no pairs with send-tag k in b^{ρ_k} we thus obtain $\langle k, \boldsymbol{f} \rangle \in b^{\rho_{k+1}} \setminus b^{\rho_k}$, which implies $\langle k, \boldsymbol{f} \rangle \in tag(k, \delta_k)$, with δ_k the set of sent messages during global transition k. Also, by definition of deliv_i, we have $R_{\text{snd}}(z, t, x_i, s_i, \bar{a}) \in \text{deliv}_i$ and thus $\langle k, \boldsymbol{f} \rangle \in m_i$. Then, by definition of α we have $\alpha(k, \boldsymbol{f}) = i$. Now by applying the induction hypothesis to $k < i$, we have $v_{\mathcal{R}}(k)[y] = v_M(z, t)[y] = u$ and thus $u \in \mu_{\mathcal{R}}(i, y)$, by definition of $\mu_{\mathcal{R}}(i, y)$. So certainly $u \leq \max(\mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$.

Thus overall $v_M(x_i, s_i)[y] \leq \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y))$.

5. We conclude $v_M(x_i, s_i)[y] = \max(\{0\} \cup \mu_{\mathcal{R}}(i, y) \cup \pi_{\mathcal{R}}(i, y)) = v_{\mathcal{R}}(i)[y]$.

□

Proposition F.27. *Let P be a Dedalus program. Let H be an input distributed database instance for P, over a network $\mathcal{N}$. Every fair stable model of pure(P) on input decl(H) is the trace of a fair run of P on input H.*

Proof. This follows from Lemma F.16, because the stable model M in this section was taken in general. □