AMALGAMATING KNOWLEDGE BASES, II: DISTRIBUTED MEDIATORS*

SIBEL ADALI
Department of Computer Science,
Institute for Advanced Computer Studies,
Institute for Systems Research,
University of Maryland,
College Park, Maryland 20742.
sibel@cs.umd.edu

and

V.S. SUBRAHMANIAN
Department of Computer Science,
Institute for Advanced Computer Studies,
Institute for Systems Research,
University of Maryland,
College Park, Maryland 20742.
vs@cs.umd.edu

Received January 19, 1994
Revised September 15, 1994

ABSTRACT

Integrating knowledge from multiple sources is an important aspect of automated reasoning systems. In [23], we presented a uniform declarative and operational framework, based on *annotated logics*, for amalgamating multiple knowledge bases and data structures (e.g. relational, object-oriented, spatial, and temporal structures) when these knowledge bases (possibly) contain inconsistencies, uncertainties, and non-monotonic modes of negation. We showed that annotated logics may be used, with some modifications, to *mediate* between different knowledge bases. The multiple knowledge bases are amalgamated by embedding the individual knowledge bases into a lattice. In this paper, we describe how, given a network of sites where the different databases reside, it is possible to define a distributed semantics for amalgamated knowledge bases. More importantly, we study how the mediator may be distributed across multiple sites so that when certain conditions are satisfied, network failures do not affect the end results of queries that a user may pose. We specify different ways of distributing the mediator to protect against different types of network link failures and develop alternative soundness and completeness results.

Keywords: heterogeneous databases, distributed computing, mediators, software integration, computational logic.

---

*This work was supported by the Army Research Office under grant number DAAL-03-92-G-0225, and by ARPA/Rome Labs contract F30602-93-C-0241 (ARPA Order Nr. A716).

1. Introduction

Integrating knowledge from multiple sources is an important aspect of automated reasoning systems. In previous work [23, 17], we presented a unifying language for integrating data/knowledge expressed across different data structures and representation paradigms, when time and uncertainty were present. The semantics of the resulting mediatory language was studied. This semantics specifies what answers a user should obtain from the system, independently of where the databases may be physically located.

In practice, however, databases are often located at different sites in a network (either a local-area network, or in a large-scale network such as the Internet). Any semantics to integrate information located across a network must address the following questions: (1) which sites can be consulted by a given site in connection with a specific query? (2) how will these sites communicate with each other? (3) what is the semantics of the integrated distributed system once answers to (1) and (2) are determined ? (4) Last, but not least, the distributed semantics must be identical to the non-distributed semantics – after all, a user should get the same answers (at least those that s/he is allowed to access, independently of where the data is located).

In this paper, we attempt to answer questions (3) and (4) above. In particular, we assume that there is some method of deciding which sites can be consulted by a given site in connection with a particular query, i.e. we make no assumptions on how this is done, only assuming that it is captured by some function. We assume a very simple communication language that conveys queries and questions across the network, but we make no claim of novelty here. Based on the answers to (1) and (2), we show how a formal semantics can be devised for the entire distributed system. Subsequently we address two points:

- first, we introduce the notion of an **acceptable placement** of mediating clauses. Then we show that any siting of clauses from the mediator which is an acceptable placement yields a soundness and completeness result, i.e. the distributed semantics will coincide with the non-distributed semantics.
- subsequently, we address the issue of **link failures**, and specify conditions under which mediating clauses can be distributed so as to guard against a fixed number (of "worst-case") of link failures in the network. The idea is that we would like the afore-mentioned completeness theorem to hold even if certain links in the network go "down." We identify some conditions under which these results hold even if links in the network go down.

The organization of the paper is as follows: in Section 2, we outline the basic ideas underlying our mediated framework [17, 23, 1]. In Section 3, we present a motivating example that will be used throughout the paper to illustrate the formal definitions. Section 4 explains the syntax and semantics of mediatory knowledge bases – this is merely a straightforward combination of [23, 17]. Section 5 defines the semantics of a distributed mediated system. In Section 6, we show how it is possible to distribute the mediator across the network so that the resulting semantics is identical to the non-distributed semantics. We also develop methods (under certain conditions) to distribute the mediator so that link failures do not affect the resulting semantics.

2. An Overview of the Syntax of Hybrid and Amalgamated Knowledge Bases

This paper is the second in a series of papers [23] developing the theory and practice of integrating information with the help of a "mediator". These papers, together with [17], uses the framework of "generalized annotated program" (GAPs for short) framework proposed in [14] to integrate information from deductive databases together with information from nonlogical databases such as relational databases, auxiliary data structures and numerical constraints. The GAP framework assumes that we have a set, $\mathcal{T}$, of truth values[1] that forms a complete lattice under an ordering $\preceq$. In this paper, we are going to use the truth value lattice $(\mathcal{UNC}, \preceq)$ which is the set of all functions from $\mathbf{R}^+$ to $[0,1]$ where $\mathbf{R}^+$ denotes the set of nonnegative reals. The ordering $\preceq$ on $\mathcal{UNC}$ is defined as follows: $f_1 \preceq f_2$ iff for all $r \in \mathbf{R}^+$, $f_1(r) \leq f_2(r)$ where $\leq$ is the usual less-than-or-equal-to ordering on the reals. For example, the expression $[0.7, \{1,2\}] \in \mathcal{UNC}$ can be viewed as the function f that assigns the truth value 0.7 to the time points 1 and 2 only, i.e. $f(1) = 0.7$, $f(2) = 0.7$ and $f(X) = 0$ for $X \notin \{1,2\}$. Hence, $[0.5, \{1,3\}] \leq [0.7, \{1,2,3\}]$, but $[0.5, \{1,3\}]$ and $[0.6, \{1,2\}]$ are incomparable. GAPs work with *annotated atoms* of the form $A : \mu$ where A is an atom (defined in the usual way) and μ is an expression whose value evaluates to a member of the truth value lattice. As an example, an annotated atom of the form $at\_robot(1,1) : [0.7, \{1,2\}]$ can be read as: at both time instants 1 and 2, there is at least a 70% certainty that the robot is at point $(1,1)$.

An annotated clause is a sentence of the form:

$$A : \mu \quad \leftarrow \quad B_1 : \mu_1 \,\&\ldots\&\, B_n : \mu_n$$

where $A, B_1, \ldots, B_n$ are atoms, and $\mu, \mu_1, \ldots, \mu_n$ are truth values. In the case of the truth value lattice $(\mathcal{UNC}, \preceq)$, each μ_i is a pair $[v_i, t_i]$ where v_i is an evaluable term denoting a real number between 0 and 1, and t_i is an evaluable term denoting a set of non-negative integers (time points).

Suppose we have a collection of deductive databases $DB_1, \ldots, DB_n$ over the lattice $(\mathcal{UNC}, \preceq)$ and a set $\Sigma_1, \ldots, \Sigma_m$ of nonlogical databases or data structures. Then the truth value lattice $(\mathcal{UNC}, \preceq)$ can be extended to $(2^{\{1,\ldots,n,\mathbf{m}\}} \times \mathcal{UNC}, \preceq)$ where $[X_1, \mu_1, t_1] \preceq [X_2, \mu_2, t_2]$ iff $X_1 \subseteq X_2$ and $[\mu_1, t_1] \leq [\mu_2, t_2]$. $\mathbf{m}$ is a special symbol referring to the "mediator" which integrates the local deductive databases $DB_1, \ldots, DB_n$. Then, an atom of the from $at\_robot(1,1) : [\{3,5\}, 0.7, \{1,2\}]$ can be read as: according to the joint information in databases DB_1 and DB_2, at both time instants 1 and 2, there is at least a 70% certainty that the robot is at point $(1,1)$. An annotated clause over this extended lattice is of the form

$$A : \mu \quad \leftarrow \quad B_1 : \mu_1 \,\&\ldots\&\, B_k : \mu_k$$

[1] It was shown in [23, 17] that the hybrid and amalgamated knowledge base framework can be easily extended to any complete truth value lattice.

where $A, B_1, \ldots, B_k$ are atoms, and $\mu, \mu_1, \ldots, \mu_k$ are truth values of the form $[D, \mu, t]$ where μ, t are as before, and D is a subset of $\{1, \ldots, n, \mathbf{m}\}$. The nonlogical databases are referred to as constraint domains. In addition to these databases, we assume there is an additional database which we call a mediator. Suppose, for instance, that we have some implementation of a spatial database that contains a pre-defined implementation of a relation $in\_room(X, Y)$ which succeeds if the point (X, Y) is inside some (fixed) room. Then, a clause of the form:

$$at\_robot(X, Y) : [\{\mathbf{m}\}, V, \mathbf{R}^{+}] \leftarrow in\_room(X, Y) \,||\, at\_robot(X, Y) : [\{1, 2\}, V, \mathbf{R}^{+}]$$

in the mediatory database can be interpreted in the following way: The mediator will conclude that the robot is at point (X, Y) with certainty V at all points in time, if databases 1 and 2 jointly assert that the robot is at this point with the same certainty and the information stored in the spatial data structure states that this point is in the room. Here the expression $in\_room(X, Y)$ is called a constraint over the spatial data structure. This constraint may be viewed as a query that is processed/evaluated by an existing implementation of the spatial data structure.

3. Motivating Example

In this section we will introduce a toy robotic example to motivate the use of distributed, heterogeneous databases. This example will serve to illustrate various concepts introduced later on in the paper.

Suppose two robots are placed in a room that contains several objects. The robots are controlled by a mediating program which issues direct commands to them and integrates information about the workspace and the properties of entities in the workspace from a variety of sources (e.g. databases of different types, different data structures, and sensor information). Such an integration may involve pooling together information from these diverse sources, and resolving conflicts between them. The mediator is distributed across several sites located on a network – furthermore, the sources being integrated by the mediator may also be located at different sites in the network. Figure 3. shows this network.

The network contains three sites, numbered 1, 2 and 3. The information available at each site reflects certain aspects of the robots' (common) workspace. Site 3 is a site that gathers information from three temperature sensors that periodically report the temperature of various objects in the workspace. The information gathered by these three sensors is contained in the databases DB_3, DB_4, DB_5, respectively. Site 3 also contains a "local" mediator that integrates this sensor information and reports, for each object, a temperature value together with an associated certainty factor. These values may change with time.

Site 2 has access to three databases. One is a relational database Σ_2 that describes static attributes of objects in the workspace. These may include the `COLOR`, `WEIGHT`, `MOBILITY`, and `DIMENSIONS` of the object. DB_1 and DB_2 are deductive databases specifying the capabilities and positional attributes of robots $r1$ and $r2$, respectively. Site 2 also has a local mediator which determines which objects a given robot can

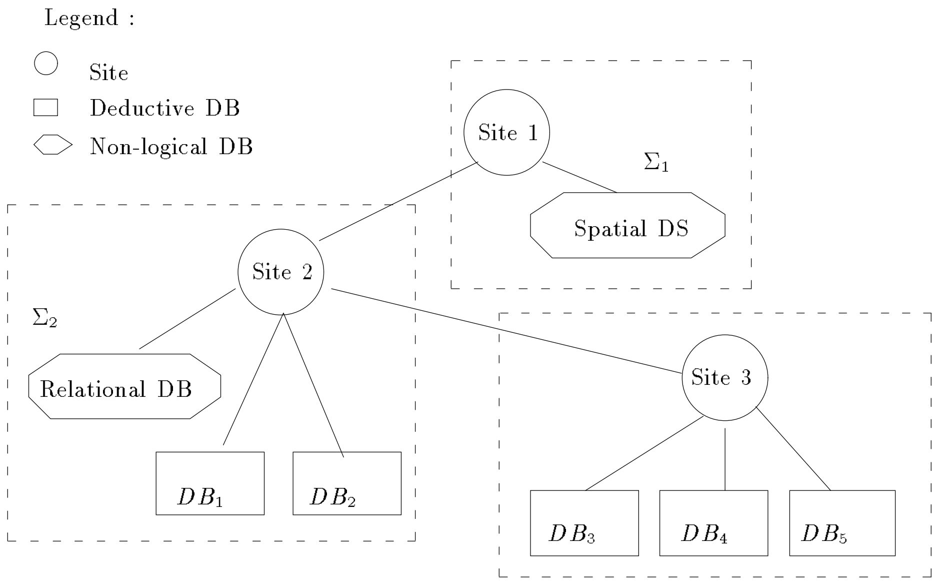

Figure 1: Distributed mediator for the robot example

move safely. However, it may need to access temperature information from Site 3. This involves making a request to a remote site.

Site 1 is connected directly to Site 2 on the network and has access to a spatial data structure Σ_1 (e.g. a quadtree) specifying the spatial layout of the workspace. In particular, this spatial data structure specifies where different objects (including the robots $r1$ and $r2$) are located. Site 1 also contains a local mediator which accesses positional information and information about robot capabilities from Site 2. Note that Site 1 may need to resolve conflicts (e.g. positional information specified by the spatial data structure may be in conflict with positional information reported by Site 2). Site 1 also uses proximity information to optimally utilize the robots. Site 1 is the "top-level" mediator, and it is responsible for eventually issuing commands to the robots.

Site 1: The workspace of the robots is a (4×4) grid, with intersection points representing the possible locations of the objects and the robots. The layout of the workspace for this example is given in figure 3. and the corresponding spatial information is stored in a data structure. There are many data structures that can be used to represent spatial data. As an example, a point quadtree (cf. Samet [20]) reflecting the spatial information about the workspace is given in figure 3.. We assume the points are inserted in the order a,b,c,d,e,f,$r1$,$r2$.

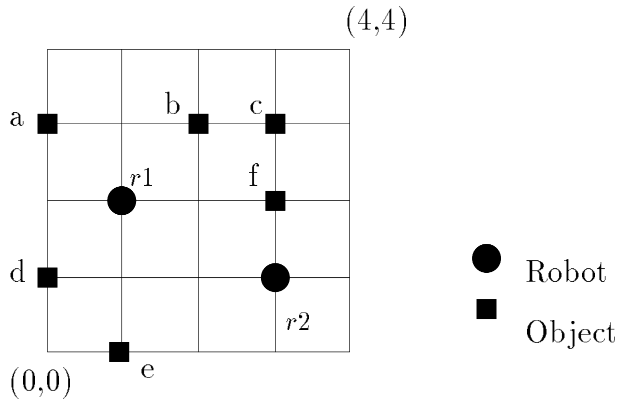

Figure 2: Robots' workspace

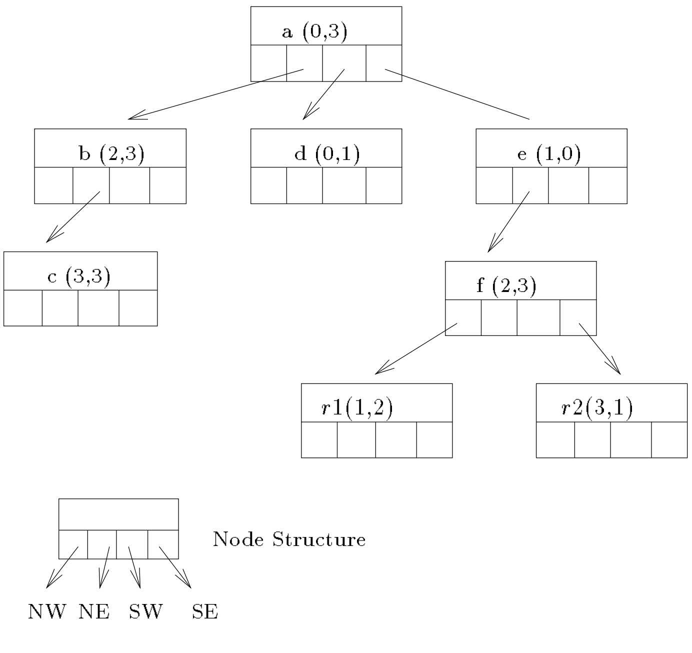

Figure 3: Point quadtree for the robot example

Features			
obj	color	weight	mobility
a	white	50	mobile
b	blue	60	mobile
c	yellow	40	immobile
d	white	30	mobile
e	black	45	mobile
f	red	100	mobile

Dimensions			
obj	width	height	length
a	10	20	20
b	15	5	20
c	10	20	20
d	10	5	5
e	10	25	10
f	40	100	50

Figure 4: Tuples stored in the relational database

Site 2: The relational database Σ_2 is located at site 2. This database is used to obtain facts about the objects. These facts correspond to those attributes of objects that don't change with time. The information stored in the database is structured in two relations, Features : $\langle$ obj,color,weight(kg.),mobility $\rangle$, Dimensions: $\langle$ obj,width(cm.),height(cm.),length(cm.) $\rangle$. The tuples stored in these two relations for the objects in the workspace is given in figure 4.

As explained above, databases DB_1 and DB_2 store information regarding the robots' capabilities and the location of the robots – the initial location and/or the current location. The robots can only move vertically or horizontally. The databases also contain information regarding the maximum speed at which each robot can move in a given direction, the maximum weight a robot can lift, the maximum values for the size and the maximum temperature of objects that the robot can handle safely. The information stored in databases DB_1 and DB_2 for robots $r1$ and $r2$ is given below:

$\boxed{DB_1:}$

$$\begin{aligned}
at(r1,1,2) : [1,\{0\}] &\quad \leftarrow \\
max_weight_capability(r1,100) : [1,\mathbf{R}^+] &\quad \leftarrow \\
max_temperature_handling(r1,65) : [1,\mathbf{R}^+] &\quad \leftarrow \\
max_distance_between_arms(r1,20) : [1,\mathbf{R}^+] &\quad \leftarrow \\
max_speed(r1,vertical,1) : [1,\mathbf{R}^+] &\quad \leftarrow \\
max_speed(r1,horizontal,2) : [1,\mathbf{R}^+] &\quad \leftarrow
\end{aligned}$$

$\boxed{DB_2:}$

$$\begin{aligned}
at(r2,3,1) : [1,\{0\}] &\quad \leftarrow \\
max_weight_capability(r2,50) : [1,\mathbf{R}^+] &\quad \leftarrow \\
max_temperature_handling(r2,90) : [1,\mathbf{R}^+] &\quad \leftarrow
\end{aligned}$$

$$
\begin{aligned}
max\_distance\_between\_arms(r2,50):[1,\mathbf{R}^{+}] &\leftarrow \\
max\_speed(r2,vertical,2):[1,\mathbf{R}^{+}] &\leftarrow \\
max\_speed(r2,horizontal,0.5):[1,\mathbf{R}^{+}] &\leftarrow
\end{aligned}
$$

Site 3: Databases DB_3,DB_4 and DB_5 contain sensor information about the temperature of the objects. Databases DB_3 and DB_4 are updated every 10 time units and the sensor updating DB_3 operates 5 time units ahead of the sensor updating DB_4. (i.e. if DB_3 is updated at time points $0, 10, 20, \ldots$, then DB_4 will be updated at time points $5, 15, 25, \ldots$.) These two sensors are considered to provide reliable information. DB_5 on the other hand, is updated every 2 time units by a very fast, but not reliable sensor. For this part of the robot example, we will assume that the sensors have completed recording information for the time points between 0 and 6 and the databases DB_3,DB_4 and DB_5 contain the following information:

$\boxed{DB_3:}$

$$
\begin{aligned}
temperature(a,45):[1,\{0\}] &\leftarrow \\
temperature(b,60):[1,\{0\}] &\leftarrow \\
temperature(c,30):[1,\{0\}] &\leftarrow \\
temperature(d,70):[1,\{0\}] &\leftarrow \\
temperature(e,43):[1,\{0\}] &\leftarrow \\
temperature(f,55):[1,\{0\}] &\leftarrow
\end{aligned}
$$

$\boxed{DB_4:}$

$$
\begin{aligned}
temperature(a,45):[1,\{5\}] &\leftarrow \\
temperature(b,65):[1,\{5\}] &\leftarrow \\
temperature(c,40):[1,\{5\}] &\leftarrow \\
temperature(d,65):[1,\{5\}] &\leftarrow \\
temperature(e,45):[1,\{5\}] &\leftarrow \\
temperature(f,55):[1,\{5\}] &\leftarrow
\end{aligned}
$$

$\boxed{DB_5:}$

$$
\begin{aligned}
temperature(a,45):[1,\{6\}] &\leftarrow \\
temperature(b,75):[1,\{6\}] &\leftarrow \\
temperature(c,45):[1,\{6\}] &\leftarrow \\
temperature(d,65):[1,\{6\}] &\leftarrow \\
temperature(e,50):[1,\{6\}] &\leftarrow \\
temperature(f,58):[1,\{6\}] &\leftarrow
\end{aligned}
$$

The above clauses can be read as follows: The atom $temperature(a,45) : [1,\{6\}]$ means that the truth value of $temperature(a,45)$ is at least 1 at time 6. Similarly, the atom $max\_weight\_capability(r2,50) : [1,\mathbf{R}^{+}]$ means that the truth value of $max\_weight\_capability(r2,50)$ is at least 1 at all time points. The language used to integrate data coming from the above sources will be described in the following section. We will then use this language to specify how the sites may draw conclusions in the presence of conflicting information.

4. Mediatory Knowledge Bases

A constraint domain [17] is a triple $\Sigma = (D, F, R)$ where D is a nonempty set, F is a set of functions (including higher order functions) on D, and R is a set of relations on D. Intuitively, the elements of D represent the data-objects we wish to reason about, the elements of F are the functions that can be applied to these data objects, and the elements of R are relationships that exist between these data objects. In [17], we showed how various heterogeneous data structures (including spatial, relational, object-oriented, etc.) can be viewed as constraint domains. As an example, consider point quadtrees[20] with nodes having an `INFO` field and `X` and `Y` fields representing coordinates. Let D be the set of all quadtrees that can be constructed using nodes having this type. Operations in F may include `RANGE(X,Y,R)` which finds all objects within `R` units of distance from `(X,Y)`, and `X_SLICE(X,Y,D)` which finds all objects in the quadtree located at $(\mathtt{X_1}, \mathtt{Y_1})$ where $||\mathtt{X} - \mathtt{X_1}|| \leq \mathtt{D}$. Relations in R may include the predicate `IN` specifying that a given node occurs in a quadtree, `SUBTREE` specifying that a given quadtree is a subtree of another, etc. Similar predicates may be defined on other kinds of quadtrees.

Definition 1 (Mediating Clause) A mediating clause is a clause of the form

$$\begin{aligned} A_0 : [\{\mathbf{m}\}, \mu_0, t_0] \quad \leftarrow \quad & (\Xi_1 \,\text{over}\, \Sigma_1) \,\&\, \ldots \,\&\, (\Xi_p \,\text{over}\, \Sigma_p) \;|| \\ & A_1 : [D_1, \mu_1, t_1] \,\&\, \ldots \,\&\, A_r : [D_r, \mu_r, t_r] \end{aligned}$$

where the μ_ℓ's, $0 \leq \ell \leq r$ are (expressions ranging over) real numbers between 0 and 1 inclusive, the t_i's, $0 \leq i \leq r$, are (expressions ranging over) sets of time points, and the Ξ_j's,$1 \leq j \leq p$, are constraints over the domain Σ_j.

Such a clause can be informally read as: If the constraints Ξ_j, $1 \leq j \leq p$, are all solvable over their respective constraint domains, and the databases in D_1 jointly assert that A_1 has truth value "at least" μ_1 at all time points in t_1 and ... the databases in D_r jointly assert that A_r has truth value "at least" μ_r at all time points in t_r, then the mediator concludes that the atom A_0 has truth value at least μ_0 at all time points in the set t_0. When the Σ_i's are clear from context, we will often simplify notation and delete the over Σ_i expressions in the above clause.

Example 1 Recall the robot example given earlier. Upto now, we have only specified the information stored in the databases. However, we haven't specified the mediating clauses located at the sites. For example, the following clause is stored at Site 2.

$$\begin{aligned} & can\_lift(r1, Obj) : [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\ & \qquad weight(Obj, W_1) \,\&\, W \geq W_1 \,\&\, width(Obj, D_1) \,\&\, D \geq D_1 \;|| \\ & \qquad max\_distance\_between\_arms(r1, D) : [\{1\}, 1, \mathbf{R}^+] \,\&\, \\ & \qquad max\_weight\_capability(r1, W) : [\{1\}, 1, \mathbf{R}^+] \,\&\, \\ & \qquad max\_temperature\_handling(r1, T) : [\{1\}, 1, \mathbf{R}^+] \,\&\, \\ & \qquad temperature(Obj, T_1) : [\{\mathbf{m}\}, 0.9, \{V_t\}] \,\&\, T_1 \leq T. \end{aligned}$$

In this example, the relational database is used to evaluate "weight" and "width" relations, and the real number constraint domain is used to evaluate the constraints "$W \geq W_1$" and "$D \geq D_1$".

Intuitively, the above clause means the following: the mediator concludes that robot $r1$ can lift the object Obj if the size and the weight of the object, as stored in the relational database, is well within the limits of the capabilities of $r1$ as given in DB_1 and the mediator knows with 90 % certainty that the temperature of the object is less than the maximum allowed value for $r1$. A similar clause is stored for the robot $r2$ as well, but the certainty factor for the temperature is needed only to be 0.8 or more. A possible reason for this may be that the temperature sensitivity of $r2$ is not a very critical variable and the upper limit stored in DB_2 is more lax.

Another example of a mediating clause is the following clause stored in Site 3:

$$temperature(a, Y) : [\{\mathbf{m}\}, V, \{V_{t_1}\}] \leftarrow temperature(a, Y) : [\{5\}, V, \{V_{t_2}\}]$$

This means that although the sensor which updates DB_5 is not very reliable, the mediator will accept whatever DB_5 says for object a, since this value is usually more recent and it may be the case that the temperature of a never exceeds the limit values the robots can handle. □

Definition 2 (Mediator) A mediator is a finite set of mediating clauses.

An atom $A : [D, \mu, t]$ is said to be *ground annotated* iff $D \subseteq \{1, \ldots, n, \mathbf{m}\}$, $\mu \in [0, 1]$ and $t \in 2^{\mathbf{R}^+}$.

Definition 3 (M-interpretation) An M-interpretation I is a mapping from the Herbrand Base B_L, of the base language, to the set of functions $f : \{1, \ldots, n, \mathbf{m}\} \rightarrow (\mathcal{UNC}, \preceq)$. That is, for all $A \in B_L$, $I(A)$ is a mapping from $\{1, \ldots, n, \mathbf{m}\}$ to $\mathcal{UNC}$.

We now extend the $\leq$ ordering to M-interpretations as follows: given two M-interpretations I_1 and I_2,

$$I_1 \leq I_2 \text{ iff } (\forall A \in B_L)(\forall D' \subseteq D) \sqcup_{d_1 \in D'} I_1(A)(d_1) \leq \sqcup_{d_2 \in D'} I_2(A)(d_2).$$

It is easy to see that the set of all M-interpretations under this ordering is a complete lattice.

Definition 4 (M-satisfaction) Suppose I is an M-interpretation, $[\mu, t] \in \mathcal{UNC}$ and $D \in \{1, \ldots, n, \mathbf{m}\}$. Then, I M-satisfies $A : [D, \mu, t]$, denoted by $I \models^{\mathrm{M}} A : [D, \mu, t]$ iff for all $t_0 \in t$, $\sqcup_{d \in D} I(A)(d)(t_0) \geq \mu$. Satisfaction of all other expressions are given in the usual way.

Example 2 Consider the following clause:

$$temperature(a, 45) : [\{\mathbf{m}\}, 0.7, \{7\}] \leftarrow temperature(a, 45) : [\{5\}, 0.7, \{6\}].$$

This clause is a ground instance of the second mediating clause given in example 1. Consider any M-interpretation I such that:

$$\begin{aligned} (I(temperature(a, 45))(5))(6) &= 0.8 \\ (I(temperature(a, 45))(\mathbf{m}))(7) &= 0.9 \end{aligned}$$

Then, the interpretation I given above M-satisfies the above clause as it satisfies both the body and the head. $\square$

Definition 5 Suppose DB_i is a GAP and $C =$

$$A_0 : [\mu_0, t_0] \leftarrow A_1 : [\mu_1, t_1] \,\&\, \ldots \,\&\, A_r : [\mu_r, t_r]$$

is an annotated clause in DB_i. Then the *mediating transform* of C, denoted $MT(C)$, is the clause:

$$A_0 : [\{i\}, \mu_0, t_0] \leftarrow A_1 : [\{i\}, \mu_1, t_1] \,\&\, \ldots \,\&\, A_r : [\{i\}, \mu_r, t_r]$$

The *mediating transform* of DB_i, denoted $MT(DB_i)$ is the set $\{MT(C) | C \in DB_i\}$.

In the appendix, we specify the mediating transform of the clauses in all databases associated with the robot example, together with a complete list of all the mediating clauses located in different sites.

Definition 6 (Mediatory Knowledge Base) Given a mediator M, a set of deductive databases $DB_1, \ldots, DB_n$ and constraint domains $\Sigma_1, \ldots, \Sigma_m$, the mediatory knowledge base Q is the set of clauses C where either C is in M or C is in the amalgamation transform of DB_i for some $1 \le i \le n$.

The amalgamation transform of a deductive database DB_i is obtained by adding the annotation $\{i\}$ to all the atoms that occur in DB_i. Hence the clause $p : [1, \{1\}] \leftarrow q : [0.5, \{1, 2\}]$ in database DB_3 will be replaced by the clause $p : [\{3\}, 1, \{1\}] \leftarrow q : [\{3\}, 0.5, \{1, 2\}]$ in the amalgamation transform of DB_3.

Definition 7 Suppose Q is a mediatory knowledge base. We associate with Q, an operator T_Q that maps M-interpretations to M-interpretations as follows: [2]

$(T'_Q(I)(A)(D))(s_0) = \sqcup\{\mu | A : [D, \mu, t] \leftarrow (\Xi_1 \text{ over } \Sigma_1) \,\&\, \ldots \,\&\, (\Xi_p \text{ over } \Sigma_p) \,\|\, B_1 : [D_1, \mu_1, t_1] \,\&\, \ldots \,\&\, B_r : [D_r, \mu_r, t_r]$ is a strictly ground instance of a clause in Q , for all $1 \le j \le p, \Sigma_j \rhd \Xi_j$, for all $1 \le i \le r, I \models^{\mathbf{M}} B_i : [D_i, \mu_i, t_i]$ and $s_0 \in t\}$.

$(T_Q(I)(A)(D))(s_0) = \sqcup_{D' \subseteq D} (T'_Q(I)(A)(D'))(s_0)$, for all $D \subseteq \{1, \ldots, n, \mathbf{m}\}$.

The upward iteration of the T_Q operator is defined in the usual way:

$$\begin{aligned}
(T_Q \uparrow 0(A)(D))(s_0) &= 0 \\
(T_Q \uparrow \alpha(A)(D))(s_0) &= T_Q((T_Q \uparrow \beta(A)(D))(s_0)) \text{ where } \alpha = \beta + 1 \\
(T_Q \uparrow \gamma(A)(D))(s_0) &= \sqcup_{\alpha < \beta}((T_Q \uparrow \alpha(A)(D))(s_0)) \text{ for limit ordinals } \gamma
\end{aligned}$$

Note that T'_Q assigns truth values to the $\mathcal{D}$-terms appearing in the head of clauses. But, if we know that $I(A)(\{1\})(1) = \mu$ and $I(A)(\{2\})(1) = \mu'$, then we can conclude that $I(A)(\{1, 2\})(1) = \mu \sqcup \mu'$. The operator T_Q is defined for this purpose. This way, the truth values of all possible $\mathcal{D}$-terms are established. The following theorem follows directly from the properties of the fixpoint operators proved in [23, 17].

[2] $\Sigma_j \rhd \Xi_j$ is true in any M-interpretation I iff the constraint Ξ_j is true in domain Σ_j.

Theorem 1 Suppose Q is any mediatory knowledge base. Then:

(i) I is an M-model of Q iff $T_Q(I) \leq I$.

(ii) If Q is negation free, then:

(a) T_Q is monotone.

(b) $Q \models^M A : [D, \mu, t]$ iff $(lfp(T_Q)(A)(D))(t_0) \leq \mu$ for all $t_0 \in t$.

(c) $lfp(T_Q) = T_Q \uparrow \gamma$ for some ordinal γ. □

The above theorem establishes a non-distributed semantics for mediatory knowledge bases. We will now develop the semantics for distributed knowledge bases and examine the conditions under which the distributed semantics is equivalent to the non-distributed semantics.

5. Distributed Mediators

We assume that there is a distributed network of sites and that the databases (deductive, spatial, relational, object-oriented, etc.) are all located at sites in this network. Mathematically, we use the word "network" to denote a graph, $\mathbf{N} = (V, E)$ where elements of V are the sites in the network, and the edges in $E \subseteq V \times V$ are the site interconnections. We assume that G is an undirected graph. We also assume that there is a set $\mathcal{D} = \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}$ of databases – the DB_i's represent deductive databases, while the Σ_j's represent non-deductive databases (constraint domains). These databases are located at various sites in the network $\mathbf{N}$. The following definition specifies this.

Definition 8 (Distribution Function) Given a network $\mathbf{N} = (V, E)$ and a set $\mathcal{D} = \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}$ of databases, a *distribution function*, $f_{\mathbf{N}}$, is a map from $\mathcal{D}$ to V.

Intuitively, $f_{\mathbf{N}}(DB_i) = v_j$ means that the database DB_i is located at site v_j in the network. $f_{\mathbf{N}}(\Sigma_j)$ is defined in a similar way.

Example 3 In the robot example, the network considered is $\mathbf{N} = (V, E)$ where $V = \{v_1, v_2, v_3\}$ and $E = \{(v_1, v_2), (v_2, v_1), (v_2, v_3), (v_3, v_2)\}$. The distribution function for this example is given by: $f_{\mathbf{N}}(DB_1) = v_2$, $f_{\mathbf{N}}(DB_2) = v_2$, $f_{\mathbf{N}}(DB_3) = v_3$, $f_{\mathbf{N}}(DB_4) = v_3$, $f_{\mathbf{N}}(DB_5) = v_3$, $f_{\mathbf{N}}(\Sigma_1) = v_1$ and $f_{\mathbf{N}}(\Sigma_2) = v_2$. □

Definition 9 (Mediatory-Distribution Function) Given a network $\mathbf{N} = (V, E)$ and a set $\mathbf{M}$ of mediatory clauses, a *mediatory distribution function*, $md_{\mathbf{N}}$ is a map from $\mathbf{M}$ to 2^V. Intuitively, if $v_i \in md_{\mathbf{N}}(C)$ for a mediatory clause C, then this means that the distributed mediator at site v_i contains the clause C. Note that a mediatory clause in $\mathbf{M}$ may be located at several sites in the network.

Example 4 Recall that in the robot example, all the mediatory clauses occur only at one site. In other words, if site i contains the clause C, then no other site contains C. The mediatory-distribution function in this case is the function which returns a singleton set containing the site at which a mediating clause is located. □

Definition 10 Given a distribution function $f_{\mathbf{N}} : \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\} \rightarrow \{v_1, \ldots, v_m\}$, the set of databases directly connected to a given site v_i is denoted by $dbs(v_i) = \{H \in \mathcal{D} \mid f_{\mathbf{N}}(H) = v_i\}$.

Note that the function dbs is the inverse of the distribution function $f_{\mathbf{N}}$. It specifies the set of databases associated with a site that can be directly queried. The information stored in $dbs(v_i)$ will be amalgamated with the mediatory clauses located in v_i to obtain a local computing environment.

Example 5 In the robot example, $dbs(v_3) = \{DB_3, DB_4, DB_5\}$, $dbs(v_2) = \{DB_1, DB_2, \Sigma_2\}$ and $dbs(v_1) = \{\Sigma_1\}$. $\square$

Definition 11 Given a network $\mathbf{N} = (V, E)$ and a distributed mediator M with a mediatory-distribution function $md_{\mathbf{N}}$, the amalgamated site knowledge base for a site v_i, denoted by $ASKB(v_i)$, is the union of the set of mediating clauses located at this site with the set of clauses obtained by applying the amalgamation transform to all the clauses in all deductive databases in $dbs(v_i)$.

Recall that the mediating transform of the clauses in the robot example is given in the appendix. The clauses located at a single site constitute the amalgamated site knowledge base for that site.

Next, we will define the concept of a distributed interpretation. Note that the standard definition of an interpretation isn't suitable for a distributed environment where sites possibly send messages to each other and exchange information. These messages will cause queries to be executed at other sites and the answers to these queries to be sent back. The following is a list of messages used in the distributed network of mediators:

- $Ask^{v_i,v_j}(A : [D, \mu, t])$ means that site v_i is asking site v_j what it knows about atom A. In other words, site v_i wants site v_j to send the answer to the query $\leftarrow A : [D, \mu, t]$.
- $Tell^{v_i,v_j}(A : [D, \mu, t])$ means that site v_i is answering the query site v_j has asked about atom A and $A : [D, \mu, t]$ is true in the distributed knowledge base according to the information available at site v_i.
- $Ask^{v_i,v_j}(\Xi \,\text{over}\, \Sigma)$ means that site v_i is asking site v_j for the solution of the constraint Ξ over the constraint domain Σ.
- $Tell^{v_i,v_j}(\Xi \,\text{over}\, \Sigma)$ means that site v_j is reporting to site v_i that Ξ is true in domain Σ.

Definition 12 (Distributed-interpretation) A distributed-interpretation $I^{\sharp}$ defined for an amalgamated knowledge base with distributed mediators and a network $\mathbf{N} = (V, E)$ is a pair $((I_{v_1}, \ldots, I_{v_m}), Msg)$ where

- I_{v_i} is an M-interpretation for $ASKB(v_i)$, and

- Msg is a set of messages X where X is one of: $Ask^{v_i,v_j}(A : [D,\mu,t])$, $Tell^{v_i,v_j}(A : [D,\mu,t])$, $Ask^{v_i,v_j}(\Xi \,\text{over}\, \Sigma)$, and $Tell^{v_i,v_j}(\Xi \,\text{over}\, \Sigma)$.

In any distributed network, there must be some governing protocol which determines when a given site asks for assistance from another site. We model this via a function, f_{Ask}, which determines, given a site v_i in the network, and an atom $A : [D,\mu,t]$, which other sites v_i should ask about $A : [D,\mu,t]$; intuitively, one may think of the sites in $f_{Ask}(v_i, A : [D,\mu,t])$ as being v_i's "friends" – those sites f_{Ask} forces v_i to consult. We say that the function f_{Ask} is *sensible* iff whenever $f_{Ask}(v_i, A : [D_1,\mu_1,t_1]) = V_{s_1}$, if $\mu_2 \le \mu_1$ and $t_2 \subseteq t_1$ and $D_2 \supseteq D_1$, and $f_{Ask}(v_i, A : [D_2,\mu_2,t_2]) = V_{s_2}$ then $V_{s_1} \subseteq V_{s_2}$. Intuitively, the condition says that if f_{Ask} consults another site about an atom $A : [D_1,\mu_1,t_1]$, and if the atom $A : [D_2,\mu_2,t_2]$ is implied by (hence weaker than) $A : [D_1,\mu_1,t_1]$, then f_{Ask} must consult the other site about $A : [D_2,\mu_2,t_2]$ as well. (We emphasize that this is a declarative requirement, and that operational procedures that implement this requirement can implicitly achieve this rather than doing so explicitly).

The function f_{Ask} is said to *preserve subsumption* iff whenever $v_j \in f_{Ask}(v_i, A : [D,\mu,t])$ and $A'\theta = A$ and $D \subseteq D'$, it is the case that $v_j \in f_{Ask}(v_i, A' : [D',\mu',t'])$. This property is similar to the sensibility property, but it is more general. For example, if site v_i believes that site v_j may know something about the atom $p(a)$, then it should also ask v_j about atom $p(X)$ since $(\forall X)p(X)$ implies $p(a)$. Similarly, if v_i believes that site v_j may contain information from databases 1 and 2 together, then v_j should also know about databases 1,2 and 3 according to v_i. Clearly, if f_{Ask} preserves subsumption then f_{Ask} is also sensible. In this way subsumption is a more general concept.

Example 6 Recall the robot example given in section 3 and the appendix. We now define a suitable query strategy function f_{Ask} for this example. For all $\mu \in [0,1]$ and $t \in 2^{\mathbf{R}^+}$ and for all the ground instances of the following atoms, the value of f_{Ask} is given as follows: ("REL"$\in \{max\_temperature\_handling(X,Y), max\_distance\_between\_arms(X,Y), max\_speed(X,Y,Z)\}$)

$$\begin{array}{rcll}
f_{Ask}(v_1, \text{"}REL\text{"} : [D,\mu,t]) &=& \{v_2\}, & D \subseteq \{1,2,\mathbf{m}\} \\
f_{Ask}(v_1, temperature(X,Y) : [D,\mu,t]) &=& \{v_2\}, & D \subseteq \{3,4,5,\mathbf{m}\} \\
f_{Ask}(v_1, at(X,Y,Z) : [D,\mu,t]) &=& \{v_2\}, & D \subseteq \{1,2,\mathbf{m}\} \\
f_{Ask}(v_1, can_lift(X,Y) : [\{\mathbf{m}\},\mu,t]) &=& \{v_2\} & \\
f_{Ask}(v_2, temperature(X,Y) : [D,\mu,t]) &=& \{v_3\}, & D \subseteq \{3,4,5,\mathbf{m}\} \\
f_{Ask}(v_2, can_lift(X,Y) : [\{\mathbf{m}\},\mu,t]) &=& \{v_1\} & \\
f_{Ask}(v_2, recent_temperature(X,Y) : [\{\mathbf{m}\},\mu,t]) &=& \{v_3\} & \\
f_{Ask}(v_2, max_possible_speed(X,Y,Z) : [\{\mathbf{m}\},\mu,t]) &=& \{v_1\} & \\
f_{Ask}(v_j, A : [D,\mu,t]) &=& \{\} & \underline{\text{for other atoms}} \\
f_{Ask}(v_1, (weight(Obj,X) \,\text{over}\, \Sigma_2 \mid Obj, X)) &=& \{v_2\} & \\
f_{Ask}(v_2, (at(Obj,X,Y) \,\text{over}\, \Sigma_2 \mid Obj, X, Y)) &=& \{v_1\} & \\
f_{Ask}(v_j, (\Xi \,\text{over}\, \Sigma)) &=& \{\} & \underline{\text{for other constraints}}
\end{array}$$

□

Definition 13 Given a network $\mathbf{N} = (V, E)$, a function f_{Ask} which determines when a site v_i asks a query about an atom A to other sites and a distributed-interpretation $I^\sharp = ((I_{v_1}, \ldots, I_{v_m}), Msg)$, the operator $F_{\mathbf{N}}$ which maps distributed-interpretations to distributed-interpretations is defined as follows.

- Let Q_{v_j} denote the amalgamation of the local databases at site v_j with the mediating clauses residing at v_j.
- Let $Q'_{v_j} = Q_{v_j} \cup \{A : [D, \mu, t] \,|\, Tell^{v_i, v_j}(A : [D, \mu, t]) \in Msg \text{ for some site } v_i\}$
- Let $I'_{v_j} = T_{Q'_{v_j}}(I_{v_j})$.
- $$\begin{aligned} Msg' = Msg &\cup \{Tell^{v_i,v_j}(A : [D,\mu,t]) \mid\ T_{Q'_{v_i}}(I_{v_i}) \models^{\mathbf{M}} A : [D,\mu,t]\ \& \\ &\qquad Ask^{v_j,v_i}(A : [D,\mu,t]) \in Msg\} \\ &\cup \{Ask^{v_i,v_j}(B : [D,\mu,t]) \mid\ T_{Q'_{v_i}}(I_i) \models^{\mathbf{M}} B : [D,\mu,t]\ \& \\ &\qquad v_j \in f_{Ask}(v_i, A : [D,\mu,t])\ \& \\ &\qquad (v_i, v_j) \in E\} \\ &\cup \{Tell^{v_i,v_j}(\Xi \text{ over } \Sigma) \mid Ask^{v_j,v_i}(\Xi \text{ over } \Sigma) \in Msg\ \&\ \Sigma \triangleright \Xi\} \\ &\cup \{Ask^{v_i,v_j}(\Xi \text{ over } \Sigma) \mid v_j \in f_{Ask}(v_i, (\Xi \text{ over } \Sigma))\ \&\ (v_i, v_j) \in E\}. \end{aligned}$$

Then $F_{\mathbf{N}}(I^\sharp) = ((I'_{v_1}, \ldots I'_{v_m}), Msg')$.

In the distributed framework, the operator $\triangleright$ is interpreted as follows: given a distributed-interpretation $I^\sharp = ((I_{v_1}, \ldots, I_{v_m}), Msg)$, $\Sigma \triangleright \Xi$ is true at site v_i iff (1) the constraint Ξ is solvable in domain Σ where Σ is located at site v_i, or (2) Msg contains a message of the form $Tell^{v_j,v_i} : (\Xi \text{ over } \Sigma)$ for some site v_j in $\mathbf{N}$.

Intuitively, the $F_{\mathbf{N}}$ operator works as follows: at every step, the local M-interpretation I_{v_i} for site v_i contains all the logical consequences of the distributed system computed so far at site v_i using the information available at this site and the information provided by other sites through the messages sent by them. In addition to that, sites have access to the global set, Msg, of messages. In practice, a site v_i will only need to know the messages of the form $\{Ask, Tell\}^{v_j,v_i}$. Then, during one execution of the $F_{\mathbf{N}}$ operator, every site executes the T_Q operator using the local M-interpretation I_{v_i}, the amalgamated site knowledge base for this site, and the messages sent to this site so far. Hence, facts of the form $A : [D, \mu, t]$ are added to the knowledge base prior to the execution of the T_Q operator for messages $Tell^{v_j,v_i}(A : [D, \mu, t])$ in Msg. During the execution of the T_Q operator, if a constraint $(\Xi \text{ over } \Sigma)$ needs to be solved and Σ is not located at site v_i, then the computation will use a relevant $Tell$ message sent to site v_i about domain Σ from the current set of messages if such a message exists. At the end of each round, the set Msg is updated as follows: every site sends a set of Ask messages for the information that can be obtained from other sites as determined by the f_{Ask} function and responds to the Ask messages sent to them using their local interpretations.

The following example illustrates the working of the operator $F_{\mathbf{N}}$. (The expression $F^i_{\mathbf{N}}$ denotes the ith iteration of the $F_{\mathbf{N}}$ operator as defined for the T_Q operator in section 4.)

Example 7 Suppose the distributed network consists of two sites only. One deductive database DB_1 and one relational database Σ_1 are located at site 1 and one deductive database DB_2 is located at site 2. Site 1 and site 2 are connected by a link. Suppose the following information is stored in the sites:

The mediator at site 1 contains the clause:

$$p(X) : [\{\mathbf{m}\}, V, T] \leftarrow \mathrm{adm}(X) \,\|\, q(X) : [\{1,2\}, V, T] \,\&\, r(X) : [\{1\}, 0.5, T].$$

The mediator at site 2 contains the clause:

$$p(X) : [\{\mathbf{m}\}, V, T] \leftarrow s(X) : [\{2\}, V, T]$$

The relational database Σ_1 contains the tuple: adm(a). Deductive databases contain the following facts:

$\boxed{DB_1 \text{ (at site 1):}}$	$\boxed{DB_2 \text{ (at site 2):}}$
$r(a) : [\{1\}, 0.7, \{1\}] \leftarrow$	$s(a) : [\{2\}, 0.6, \{1,2\}] \leftarrow$
$r(b) : [\{1\}, 0.5, \{1\}] \leftarrow$	$s(b) : [\{2\}, 0.5, \{1,2\}] \leftarrow$
$q(a) : [\{1\}, 0.3, \{1\}] \leftarrow$	$q(X) : [\{2\}, 0.7, T] \leftarrow t(X) : [\{2\}, 0.5, T]$
$q(b) : [\{1\}, 0.8, \{1\}] \leftarrow$	$t(X) : [\{2\}, 0.5, T] \leftarrow s(X) : [\{2\}, 0.5, T]$

Suppose that $f_{Ask}(v_1, A : [D, \mu, t]) = \{v_2\}$ for all $A : [D, \mu, t]$ appearing in mediating clauses in site v_1 and $f_{Ask}(v_2, A : [D, \mu, t]) = \{\}$ for all $A : [D, \mu, t]$ appearing in v_2. Also suppose that $f_{Ask}(v_1, (adm(a))) = f_{Ask}(v_2, (adm(a))) = \{\}$. Since the sites never ask about the predicates q, r, s and t, the truth value of these predicates is determined at the local sites.

Let $I^\sharp = ((I_\perp, I_\perp), \emptyset)$ where $I_\perp$ is the interpretation that assigns $\perp$ (0 in the case of $\mathcal{UNC}$) to all the atoms in the underlying language. Then, for all $t \in \{1, 2\}$ and for all $i \leq 2$, $F^i_{\mathbf{N}}(I^\sharp) = ((I^i_{v_1}, I^i_{v_2}), Msg^i)$ assigns the following truth values:

$$\begin{array}{lcl}
(I^i_{v_1}(r(a)(1)))(1) & = & 0.7 \\
(I^i_{v_1}(r(b)(1)))(1) & = & 0.5 \\
(I^i_{v_1}(q(a)(1)))(1) & = & 0.3 \\
(I^i_{v_1}(q(b)(1)))(1) & = & 0.8 \\
(I^i_{v_2}(q(a)(2)))(t) & = & 0
\end{array}
\qquad
\begin{array}{lcl}
(I^i_{v_2}(s(a)(2)))(t) & = & 0.6 \\
(I^i_{v_2}(s(b)(2)))(t) & = & 0.5 \\
(I^i_{v_2}(t(a)(2)))(t) & = & 0.5 \\
(I^i_{v_2}(t(b)(2)))(t) & = & 0.5 \\
(I^i_{v_2}(q(b)(2)))(t) & = & 0
\end{array}$$

For $i \geq 3$, the above equalities hold except in the two cases listed below:

$$\begin{array}{lcl}
(I^i_{v_2}(q(a)(2)))(t) & = & 0.7 \\
(I^i_{v_2}(q(b)(2)))(t) & = & 0.7
\end{array}$$

The truth values assigned to the atom $p(X) : [\{\mathbf{m}\}, V, t]$ for all time points $t \in \{1, 2\}$ by the interpretations $I^i_{v_1}$ and $I^i_{v_2}$ under different substitutions are given in the table below. The new messages that affect the truth values assigned to this atom are also given at each step.

	$I^i_{v_1}$		$I^i_{v_2}$		
$p(X)$	$X=a$	$X=b$	$X=a$	$X=b$	New Messages
$i=0$	0	0	0	0	$\emptyset$
$i=1$	0.3	0	0.6	0.5	$Ask^{v_1,v_2}(p(X):[\{\mathbf{m}\},V,T])$ $Ask^{v_1,v_2}(q(X):[\{1,2\},V,T])$
$i=2$	0.3	0	0.6	0.4	$Tell^{v_2,v_1}(p(a):[\{\mathbf{m}\},0.6,\{1,2\}])$ $Tell^{v_2,v_1}(p(b):[\{\mathbf{m}\},0.5,\{1,2\}])$ $Tell^{v_2,v_1}(q(X):[\{1,2\},0,\{1,2\}])$
$i=3$	0.6	0.4	0.6	0.4	$Tell^{v_2,v_1}(q(X):[\{1,2\},0.7,\{1,2\}])$
$i\geq 4$	0.7	0.4	0.6	0.4	$\emptyset$

$\square$

We now extend the $\leq$ ordering defined on M-interpretations to distributed interpretations. Recall that the distributed knowledge about an atom A for a given site v_j is obtained from the local interpretation for this site as well as from messages of the form $Tell^{v_i,v_j}(A)$ that constitute the that answers site v_j has obtained to its queries about atom A. Hence, given a network $\mathbf{N}=(V,E)$ and two distributed-interpretations $I_1^\sharp=((I_{v_{1_1}},\ldots,I_{v_{m_1}}),Msg_1)$ and $I_2^\sharp=((I_{v_{1_2}},\ldots,I_{v_{m_2}}),Msg_2)$, we say that $I_1^\sharp\leq I_2^\sharp$ iff $Msg_1\subseteq Msg_2$ and $I_{v_{j_1}}\leq I_{v_{j_2}}$ for all $1\leq j\leq m$.

Definition 14 (Negation-Free Network) A network $\mathbf{N}=(V,E)$ is called *negation-free* iff the amalgamated site knowledge bases Q_{v_i} are negation-free for all sites $v_i\in V$.

Theorem 2 Let $\mathbf{N}=(V,E)$ be a negation-free network. Then $F_{\mathbf{N}}$ is monotone.

Proof: Suppose $\mathbf{N}=(V,E)$ is a network, and $I^\sharp$ and $J^\sharp$ are distributed-interpretations such that $I^\sharp\leq J^\sharp$. Let $I^\sharp=((I_{v_1},\ldots,I_{v_m}),Msg_1)$ and $J^\sharp=((J_{v_1},\ldots,J_{v_m}),Msg_2)$. We need to show that $F_{\mathbf{N}}(I^\sharp)\leq F_{\mathbf{N}}(J^\sharp)$. Let $F_{\mathbf{N}}(I^\sharp)=((I^\odot_{v_1},\ldots,I^\odot_{v_m}),Msg^\odot)$ and $F_{\mathbf{N}}(J^\sharp)=((J^\square_{v_1},\ldots,J^\square_{v_m}),Msg^\square)$. We will first show that $I^\odot_{v_\ell}\leq J^\square_{v_\ell}$ for all $1\leq\ell\leq m$.

$I^\odot_{v_\ell}=T_{Q'_{v_\ell}}(I_{v_\ell})$. Likewise, $J^\square_{v_\ell}=T_{Q''_{v_\ell}}(J_{v_\ell})$ where

$$\begin{aligned}
Q'_{v_\ell} &= Q_{v_\ell}\cup\{A:[D,\mu,t]\,|\,Tell^{v_i,v_\ell}(A:[D,\mu,t])\in Msg_1 \text{ for some site } v_i\}\\
Q''_{v_\ell} &= Q_{v_\ell}\cup\{A:[D,\mu,t]\,|\,Tell^{v_i,v_\ell}(A:[D,\mu,t])\in Msg_2 \text{ for some site } v_i\}.
\end{aligned}$$

As $Msg_1\subseteq Msg_2$ by virtue of the fact that $I^\sharp\leq J^\sharp$, it follows that $Q'_{v_\ell}\subseteq Q''_{v_\ell}$. We know, by monotonicity of the T_Q operator as proved in 1, that $T_{Q'_{v_\ell}}(I^\odot_{v_\ell})\leq T_{Q'_{v_\ell}}(J^\square_{v_\ell})$. As $Q'_{v_\ell}\subseteq Q''_{v_\ell}$ and both Q'_{v_ℓ} and Q''_{v_ℓ} are negation-free, it follows that $T_{Q'_{v_\ell}}(J^\square_{v_\ell})\leq T_{Q''_{v_\ell}}(J^\square_{v_\ell})$, and hence, $T_{Q'_{v_\ell}}(I^\odot_{v_\ell})\leq T_{Q''_{v_\ell}}(J^\square_{v_\ell})$.

It only remains to show that $Msg^{\odot} \subseteq Msg^{\Box}$. Any formula in $Msg^{\odot}$ must be either of the form $Tell^{v_i,v_j}(A : [D, \mu, t])$ or $Ask^{v_i,v_j}(A : [D, \mu, t])$. We consider the first case – the second case is entirely symmetric.

If $Tell^{v_i,v_j}(A : [D, \mu, t]) \in Msg^{\odot}$, then one of the following two cases is applicable.

Case 1: $Tell^{v_i,v_j}(A : [D, \mu, t]) \in Msg_1$. In this case, as $I^\sharp \leq J^\sharp$, we know that $Msg_1 \subseteq Msg_2$ and hence, $Tell^{v_i,v_j}(A : [D, \mu, t]) \in Msg_2 \subseteq Msg_2^{\Box}$.

Case 2: In this case, $A : [D, \mu, t]$ is M-satisfied by $T_{Q'}(I_{v_i})$ and $Ask^{v_j,v_i}(A : [D, \mu, t]) \in Msg_1$. (Here, Q' is constructed as articulated earlier in the proof). Then, as $Msg_1 \subseteq Msg_2$, it follows that $Ask^{v_j,v_i}(A : [D, \mu, t]) \in Msg_2$. Furthermore, as $I_{v_i} \leq J_{v_i}$, it follows, by the argument above, that $T_{Q'}(I_{v_i}) \leq T_{Q''}(J_{v_i})$. Hence, $T_{Q''}(J_{v_i})$ M-satisfies $A : [D, \mu, t]$. It follows that $Tell^{v_i,v_j}(A : [D, \mu, t]) \in Msg^{\Box}$.

If $Tell^{v_i,v_j}(\Xi \,\mathrm{over}\, \Sigma) \in Msg^{\odot}$, then there are two cases to consider. These cases are similar to the cases above. Showing that all atoms of the form $Ask^{v_i,v_j}(A : [D, \mu, t])$ and the form $Ask^{v_i,v_j}(\Xi \,\mathrm{over} \Sigma)$ that are in $Msg^{\odot}$ are also in $Msg^{\Box}$ is symmetric. $\Box$

The above result immediately allows us to conclude that the operator $F_{\mathbf{N}}$ has a least fixpoint. The semantics of programs (both imperative and logical) have long been characterized by the least fixpoints of associated operators (cf. Manna [18]), and hence, we will consider this least fixpoint of F_N to be the meaning of the distributed network of databases. We will subsequently show (cf. Lemmas 2 and 1) that this least fixpoint is a generalization of the semantics for amalgamating knowledge bases proposed in [23, 17]. Those semantics have a clearly defined model-theoretic basis. Defining a model-theoretic basis for a network of databases is related to database updates because messages received by a database from another database needs to be assimilated and can be viewed as an update. Studying the semantics of updates is beyond the scope of the current paper.

Corollary 1 Suppose $\mathbf{N} = (V, E)$ is a negation-free network. Then the function $F_{\mathbf{N}}$ has a least fixpoint, denoted $\mathsf{lfp}(F_{\mathbf{N}})$. $\Box$

Corollary 2 Suppose $I^\sharp$ is any distributed interpretation, $\mathbf{N}$ is a network, and f_{Ask} is sensible. Let $F_{\mathbf{N}}(I^\sharp) = ((I_{v_1}, \ldots, I_{v_m}), Msg)$. Then Msg is closed under consequence in the following sense:

(i) if $Ask^{v_i,v_j}(A : [D_1, \mu_1, t_1]) \in Msg$, then $Ask^{v_i,v_j}(A : D_2, \mu_2, t_2) \in Msg$ for all $\mu_2 \leq \mu_1$, $t_2 \subseteq t_1$ and $D_2 \supseteq D_1$.

(ii) if $Tell^{v_i,v_j}(A : [D_1, \mu_1, t_1]) \in Msg$, then $Tell^{v_i,v_j}(A : [D_2, \mu_2, t_2]) \in Msg$ for all $\mu_2 \leq \mu_1$, $t_2 \subseteq t_1$ and $D_2 \supseteq D_1$.

Proof. (1) If $Ask^{v_i,v_j}(A : [D_1, \mu_1, t_1] \in Msg$, then $T_{Q_{v_i}}(I_{v_i})$ M-satisfies $A : [D_1, \mu_1, t_1]$ and $v_j \in f_{Ask}(v_i, A : [D_1, \mu_1, t_1])$. Hence, $T_{Q_{v_i}}(I_{v_i})$ M-satisfies $A : [D_2, \mu_2, t_2]$ as $\mu_2 \leq \mu_1$, $t_2 \subseteq t_1$ and $D_2 \supseteq D_1$. As f_{Ask} is sensible,$f_{Ask}(v_i, A : [D_1, \mu_1, t_1]) \subseteq f_{Ask}(v_i, A : [D_2, \mu_2, t_2])$, hence $v_j \in f_{Ask}(v_i, A : [D_2, \mu_2, t_2])$, the result follows immediately.

(2) The proof of part (2) is similar. Suppose $Tell^{v_i,v_j}(A:[D_1,\mu_1,t_1]) \in Msg$. Then $A:[D_1,\mu_1,t_1]$ is M-satisfied by $T_{Q'_{v_i}}(I_{v_i})$ where these notations are defined as in Definition 13. Thus, $A:[D_2,\mu_2,t_2]$ is M-satisfied by $T_{Q'_{v_i}}(I_{v_i})$ as $\mu_2 \leq \mu_1$, $t_2 \subseteq t_1$ and $D_2 \supseteq D_1$. By Part (2), it follows also that as $Ask^{v_j,v_i}(A:[D_1,\mu_1,t_1]) \in Msg$, $Ask^{v_j,v_i}(A:[D_2,\mu_2,t_2]) \in Msg$. This completes the proof. $\square$

Though message lists are closed under consequence when defining the least fixpoint of $F_{\mathbf{N}}$, in an implementation, an explicit listing is not required. The reason is that an atom $A:[D,\mu,t]$ may be used to represent all the atoms that are implied by $A:[D,\mu,t]$ and this (potentially large) set is thus captured by just a single atom.

The $F_{\mathbf{N}}$ operator defined on the distributed network is a straightforward generalization of the A_Q operator defined in [23] as well as the T_{DB} operator defined in [17].

Lemma 1 Suppose $\mathbf{N} = (\{v\},\emptyset)$ is a network consisting of just one site, and suppose $\mathcal{D} = \{DB, \Sigma_1,\ldots,\Sigma_m\}$ are the databases located at this site, where DB is a deductive database and Σ_i is a non-logical database for $1 \leq i \leq m$. Let Q_v be the amalgamated site knowledge base for site v. Then, given any distributed interpretation $I^\sharp$ such that $I^\sharp = ((I_v),\emptyset)$, if $F_{\mathbf{N}}(I^\sharp) = ((I_v^\square), Msg)$ then $I_v^\square = T_{DB}(I_v)$ where T_{DB} is the fixpoint operator defined for the hybrid knowledge bases in [17] applied for Q_v.

Proof. Let $I^\sharp = ((I_v),\emptyset)$ and Q_v be given as above. Suppose $F_{\mathbf{N}}(I^\sharp) = ((I_v^\square), Msg)$. Since there are no messages in $I^\sharp$, $Q'_v = Q_v$. Then, $I_v^\square = T_{Q_v}(I_v)$. Since, v contains only one deductive database, the definition of T_{Q_v} can be simplified to:
$(T_{Q_v}(I)(A))(s_0) = \sqcup\{\mu | A : [\mu,t] \leftarrow (\Xi_1 \,\text{over}\, \Sigma_1) \,\&\, \ldots \,\&\, (\Xi_p \,\text{over}\, \Sigma_p) \mid\mid B_1 : [\mu_1,t_1] \,\&\ldots\&\, B_r : [\mu_r,t_r]$ is a strictly ground instance of a clause in Q_v, for all $1 \leq j \leq p, \Sigma_j \rhd \Xi_j$, for all $1 \leq i \leq r, I \models^{\mathrm{M}} B_i : [\mu_i,t_i]$ and $s_0 \in t\}$.

Similarly, the definition of M-satisfaction is simplified : $I \models^{\mathrm{M}} A : [\mu,t]$ iff $I(A)(t_0) \geq \mu$ for all $t_0 \in t$. This definition of T_{Q_v} is identical to the definition of T_{DB} given in [17], hence $I_v^\square = T_{DB}(I_v)$. $\square$

The following lemma is proved along analogous lines.

Lemma 2 Suppose $\mathbf{N} = (\{v\},\emptyset)$ is a network consisting of just one site, and suppose $\mathcal{D} = \{DB_1,\ldots,DB_n\}$ are deductive databases located at this site, i.e. $f_{\mathbf{N}}(DB_i) = v$ for $1 \leq i \leq n$. Let Q_v be the amalgamated site knowledge base for site v. Given any distributed interpretation $I^\sharp$ such that $I^\sharp = ((I_v),\emptyset)$, if $F_{\mathbf{N}}(I^\sharp) = ((I_v^\square), Msg)$ then $I_v^\square = A_Q(I_v)$ where A_Q is the fixpoint operator defined for the amalgamated knowledge bases in [23]. $\square$

As we have seen so far in this section, when a set of databases is distributed across a network, the inferences made by the system depend on several factors – these are:

- the network $\mathbf{N} = (V,E)$,

- the databases $\{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}$,
- the distribution function $f_{\mathbf{N}}$,
- the function f_{Ask},
- the mediatory distribution function $md_{\mathbf{N}}$,
- the set $\mathbf{M}$ of mediating clauses.

We will refer to the 7-tuple

$$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

as a *distributed mediated system.*

Observe that for any network $\mathbf{N}$, the definition of the operator $F_{\mathbf{N}}$ defined thus far actually uses all components of a distributed mediated system. Hence, it is just as appropriate to associate the operator $F_{\mathbf{N}}$ with a distributed mediated system and to denote it by $F_{\mathtt{DMS}}$, with the same meaning and definition as $F_{\mathbf{N}}$.

6. Mediatory Distribution Functions

In this section, we study how to distribute the mediating clauses in $\mathbf{M}$ across the different sites in the network so that the resulting distributed semantics achieves the same effect as it would if $\mathbf{M}$ were completely stored at all sites in the network. In technical terms, suppose

$$\mathtt{DMS}_0 = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is a distributed mediated system where $\mathbf{N} = (V, E)$ and such that $md_{\mathbf{N}}(v) = \mathbf{M}$ for all $v \in V$, i.e. the mediator $\mathbf{M}$ is "completely" stored at all sites in the network. We are looking for:

- a characterization of a mediatory distribution function, $md'_{\mathbf{N}}$, such that the distributed mediated system

  $$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md'_{\mathbf{N}}, \mathbf{M})$$

  has the same the least fixpoint as $\mathtt{DMS}_0$ (i.e. has the same semantics as $\mathtt{DMS}_0$), and

- a characterization that preserves the same least fixpoint when various arcs in the network are allowed to go "down", i.e. if $X \subseteq E$, and

  $$\mathtt{DMS}' = (V, X, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md'_{\mathbf{N}}, \mathbf{M})$$

  then the least fixpoint of $F_{\mathtt{DMS}'}$ coincides with the least fixpoint of $F_{\mathtt{DMS}_0}$. Here, the arcs in $(E - X)$ are the ones that "go down."

We first consider the case when all links in the network are assumed to function properly, i.e. no links "go down." Subsequently, we will consider the situation when link failures occur.

Suppose we consider

$$\texttt{DMS}_0 = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

and suppose $v_i \in V$ is a specific site. Suppose site v_i needs to acquire information about the atom $A : [D, \mu, t]$. Then this information can be obtained by consulting relevant sites in the network. The following definition specifies the set of sites that may be queried (directly or indirectly) in connection with a particular atom.

Definition 15 Suppose DMS is a distributed mediated system. The *access set* of site v_i w.r.t. atom $A : [D, \mu, t]$, denoted $\texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$ is defined as follows:

(i) $v_i \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$

(ii) if $v_j \in f_{Ask}(v_i, A : [D, \mu, t])$ and $(v_i, v_j) \in E$, then $v_j \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$ and furthermore, $\texttt{ACCESS}_{\texttt{DMS}}(v_j, A : [D, \mu, t]) \subseteq \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$.

(iii) Nothing else is in $\texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$.

Intuitively, condition (1) above specifies that the set of sites v_i can turn to for help (either directly or indirectly) in relation to the atom $A : [D, \mu, t]$ includes v_i itself. The first half of condition (2) says that v_i may also ask any site v_j that is deemed by v_i to be knowledgeable about $A : [D, \mu, t]$ (i.e. $v_j \in f_{Ask}(v_i, A : [D, \mu, t])$). The second half of Condition (2) says that if v_i can turn to a site v_j for help (as above), and if site v_j is allowed to access site v_k in connection with the atom $A : [D, \mu, t]$, then v_k is in site v_i's access set w.r.t. the atom $A : [D, \mu, t]$.

Example 8 Recall the robot example given in the appendix. Let $\texttt{DMS}_R$ be the distributed mediated system for this example.

$$\begin{aligned}\texttt{DMS}_R &= (\{v_1, v_2, v_3\}, \{(v_1, v_2), (v_2, v_1), (v_2, v_3), (v_3, v_2)\}, \{DB_1, \ldots, DB_5, \Sigma_1, \Sigma_2\},\\ &\quad f_{\mathbf{N}}, md_{\mathbf{N}}, \mathbf{M})\end{aligned}$$

Definitions of $f_{\mathbf{N}}$ and $md_{\mathbf{N}}$ for this example were given in examples 3 and 4. A complete list of all the mediating clauses $\mathbf{M}$ was given in the appendix. Finally let f_{Ask} be defined as in example 6. Then the following is true for $\texttt{DMS}_R$:

$$\begin{aligned}
\texttt{ACCESS}_{\texttt{DMS}_R}(v_3, temperature(X, Y) : [D, \mu, t]) &= \{v_3\},\\
&\quad D \subseteq \{\mathbf{m}, 3, 4, 5\}\\
\texttt{ACCESS}_{\texttt{DMS}_R}(v_2, temperature(X, Y) : [D, \mu, t]) &= \{v_2, v_3\},\\
&\quad D \subseteq \{\mathbf{m}, 3, 4, 5\}\\
\texttt{ACCESS}_{\texttt{DMS}_R}(v_1, temperature(X, Y) : [D, \mu, t]) &= \{v_1, v_2, v_3\},\\
&\quad D \subseteq \{\mathbf{m}, 3, 4, 5\}\\
\texttt{ACCESS}_{\texttt{DMS}_R}(v_2, recent_temperature(X, Y) : [\{\mathbf{m}\}, \mu, t]) &= \{v_2, v_3\}
\end{aligned}$$

$$\begin{aligned}
\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_1, recent_temperature(X,Y) : [\{\mathbf{m}\}, \mu, t]) &= \{v_1\} \\
\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_1, can_lift(X,Y) : [\{\mathbf{m}\}, \mu, t]) &= \{v_1, v_2\} \\
\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_2, can_lift(X,Y) : [\{\mathbf{m}\}, \mu, t]) &= \{v_1, v_2\} \\
\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_2, max_speed(X,Y,Z) : [\{1,2\}, \mu, t]) &= \{v_2\} \\
\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_1, max_speed(X,Y,Z) : [\{1,2\}, \mu, t]) &= \{v_1\}
\end{aligned}$$

□

6.1.1. Acceptable Placements

We now specify what constitutes an *acceptable placement* of a mediating clause. Intuitively, suppose $C \in \mathbf{M}$ is a mediating clause of the form:

$$\begin{aligned}
A_0 : [\{\mathbf{m}\}, \mu_0, t_0] \quad \leftarrow \quad & (\Xi_1 \,\mathrm{over}\, \Sigma_1) \,\&\, \ldots \,\&\, (\Xi_p \,\mathrm{over}\, \Sigma_p) \,\| \\
& A_1 : [D_1, \mu_1, t_1] \,\&\ldots\&\, A_r : [D_r, \mu_r, t_r].
\end{aligned}$$

Then clearly we want C to be placed at a set of sites such that all subgoals occurring in its body are accessible to the sites at which C is placed.

Definition 16 Suppose C is a mediating clause of the above form and

$$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is a distributed mediated system. An *acceptable placement* of C is a set $X \subseteq V$ satisfying the following conditions:

(i) for all $1 \leq i \leq r$, if $ASKB(v_j)$ contains a clause having head $A_i^* : [D_i^*, \mu_i^*, t_i^*]$ such that A_i and A_i^* are unifiable via mgu θ and $D_i^* \subseteq D_i$, then

$$\mathtt{ACCESS}_{\mathtt{DMS}}(v_j, A_i^* : [D_i^*, \mu_i^*, t_i^*]\theta) \subseteq \bigcup_{v \in X} \mathtt{ACCESS}_{\mathtt{DMS}}(v, A_i : [D_i, \mu_i, t_i]).$$

(ii) for all $1 \leq w \leq p$, there exists an integer ℓ_w such that there is a path $v_i, v_{i+1}, \ldots, v_{i+k} = v_{\ell_w}$ in (V, E) such that

  (a) $v_i \in X$, and

  (b) $\Sigma_w \in dbs(v_{\ell_w})$, and

  (c) $v_{i+r} \in f_{Ask}(v_{i+r-1}, (\Xi_i \,\mathrm{over}\, \Sigma_i))$ for all $1 \leq r \leq k$.

(iii) No strict subset of X satisfies the above two conditions.

A set X that satisfies conditions (i) and (ii) above (but not necessarily (iii)) is called a *semi-placement* of C.

Intuitively, an acceptable placement is a set of sites at which C can be located. Note that this means that each and every site in X must have C located in it. Condition (1) in the above definition says that for a particular set to be considered an acceptable placement of C, it must be the case that all sites having clauses that have in their head, an atom that possibly contributes to solving a subgoal of C (i.e. an atom in C's body) must be accessible to some site in X. Condition (2) says that all constraint domains that C may need to ask for assistance must be accessible as well. Condition (3) says simply that we do not want to place C at more places than are strictly required.

Given a distributed mediated system DMS and a mediating clause C, if there exists an acceptable placement of C, then placing this clause at every site is certainly acceptable. Another greedy placement strategy that is not optimal would be to place C at every site v_i such that v_i either contains a clause whose head is unifiable with an atom in the body of C, or a domain Σ accessed by C is located at v_i. Finally, another algorithm would be to place C initially at a site v_i where v_i contains a set of clauses (similarly for domains) that are unifiable with an atom in the body of C. Then, calculate the ACCESS sets from this site, mark the sites that still need to be accessed, pick one from this set and continue until there is no such site left. It is possible, of course, that there is no acceptable placement for C in DMS. The above algorithms guarantee that an acceptable placement will be found, if one exists. Hence, the acceptability of the placements found by these algorithms should be checked at the final stage.

It can easily be seen that for *strong* mediating clauses (clauses that only have the $\{\mathbf{m}\}$ annotation in the head) of the form

$$\begin{aligned} A_0 : [\{\mathbf{m}\}, \mu_0, t_0] \quad \leftarrow \quad & (\Xi_1 \text{ over } \Sigma_1) \,\&\ldots\&\, (\Xi_p \text{ over } \Sigma_p) \,\| \\ & A_1 : [D_1, \mu_1, t_1] \,\&\ldots\&\, A_r : [D_r, \mu_r, t_r]. \end{aligned}$$

where each D_j, $1 \leq j \leq r$, is a subset of $\{1, \ldots, n\}$, it suffices to determine the placements of clauses in $\mathbf{M}$ one by one. In particular, the D_j's are not allowed to evaluate to a set with $\mathbf{m}$ in it. Consequently, such clauses never refer to other mediating clauses in their body. The following result shows that if we take two mediatory distribution functions $md^1_{\mathbf{N}}$ and $md^2_{\mathbf{N}}$ such that for all clauses $C \in \mathbf{M}$, $md^1_{\mathbf{N}}(C) \subseteq md^2_{\mathbf{N}}(C)$, then it is the case that least fixpoint of $F_{\mathtt{DMS}_1}$ is less than (according to the $\leq$-ordering) than the least fixpoint of $F_{\mathtt{DMS}_2}$ where $\mathtt{DMS}_1$ and $\mathtt{DMS}_2$ are identical to each other except that they differ on $md^1_{\mathbf{N}}$ and $md^2_{\mathbf{N}}$.

Theorem 3 (Monotonicity w.r.t. Mediatory Distribution Functions) Suppose $\mathbf{M}$ is a set of strong mediating clauses, and suppose

$$\mathtt{DMS}_1 = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md^1_{\mathbf{N}}, \mathbf{M})$$

and

$$\mathtt{DMS}_2 = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md^2_{\mathbf{N}}, \mathbf{M}).$$

Furthermore, suppose that $md^1_{\mathbf{N}}(C) \subseteq md^2_{\mathbf{N}}(C)$ for all $C \in \mathbf{M}$. Let $I^\sharp$ be any distributed interpretation such that $I^\sharp = ((I_{v_1}, \ldots, I_{v_m}), Msg)$. Suppose $F_{\mathtt{DMS}_1}(I^\sharp) = ((J_{v_1}, \ldots, J_{v_m}), Msg_1)$ and $F_{\mathtt{DMS}_2}(I^\sharp) = ((H_{v_1}, \ldots, H_{v_m}), Msg_2)$. Then,

(i) for all $1 \leq i \leq m$, $J_{v_i} \leq H_{v_i}$, and

(ii) $Msg_1 \subseteq Msg_2$.

Proof. Suppose $J_{v_i}(A)(D)(s) = \mu_1$ and $H_{v_i}(A)(D)(s) = \mu_2$. Let us examine the definition of the F operator (Definition 13).

Let $Q^\ell_{v_i}$ denote the amalgamation of the local databases with the mediating clauses residing at site v_i according to $md^\ell_{\mathbf{N}}$ together with the facts obtained from the messages of the form $Tell^{v_j,v_i}(A) \in Msg$ as explained in definition 13. First observe that $Q^1_{v_i} \subseteq Q^2_{v_i}$ because $md^1_{\mathbf{N}}(C) \subseteq md^2_{\mathbf{N}}(C)$ for all $C \in \mathbf{M}$; hence, if clause $C \in \mathbf{M}$ is at site v_i according to $\mathtt{DMS}_1$, then C must be at site v_i according to $\mathtt{DMS}_2$ as well.

Let $Msg_\ell = Msg \cup \{Tell^{v_i,v_j}(A : [D, \mu, t]) \mid A : [D, \mu, t]$ is M-satisfied by $T_{Q^\ell_{v_i}}(I_{v_i})$ and $Ask^{v_j,v_i}(A : [D, \mu, t]) \in Msg\} \cup \{Ask^{v_i,v_j}(B : [D, \mu, t]) \mid T_{Q^\ell_{v_i}}(I_{v_i})$ M-satisfies $B : [D, \mu, t]$ and $v_j \in f_{Ask}(v_i, A : [D, \mu, t])\} \cup$ constraint messages as given in definition 13. Consider $\ell = 1, 2$. It is easy to see that the first component of the above union when $\ell = 1$ is a subset of the first component when $\ell = 2$ because of the monotonicity of $T_{Q^\ell_{v_i}}$ proved in [23] and because $Q^1_{v_i} \subseteq Q^2_{v_i}$. The same observation holds for the second component of the union. The part of the constraint messages do not depend on ℓ, hence it is identical. This completes the proof that $Msg_1 \subseteq Msg_2$.

Now observe that $J_{v_i} = T_{Q^1_{v_i}}(I_{v_i})$ and $H_{v_i} = T_{Q^2_{v_i}}(I_{v_i})$. As $Q^1_{v_i} \subseteq Q^2_{v_i}$, it follows that

$$J_{v_i} = T_{Q^1_{v_i}}(I_{v_i}) \subseteq T_{Q^2_{v_i}}(I_{v_i}) = H_{v_i}.$$

This completes the proof. □

The following theorem shows that as long as all mediating clauses are placed at all sites in an acceptable placement, the resulting distributed semantics corresponds to the naive semantics.

Theorem 4 (Soundness and Completeness of Acceptable Placement Distribution Strategy) Suppose $\mathbf{M}$ is a set of strong mediating clauses, and suppose

$$\mathtt{DMS}_0 = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is such that $md_{\mathbf{N}}(C) = V$ for all $C \in \mathbf{M}$. Let $md'_{\mathbf{N}}$ be any mediating distribution function such that for all $C \in \mathbf{M}$, $md'_{\mathbf{N}}(C)$ is an acceptable placement for C. Let

$$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_k\}, f_{\mathbf{N}}, f_{Ask}, md'_{\mathbf{N}}, \mathbf{M}).$$

Then the least fixpoint of $F_{\mathtt{DMS}_0}$ is identical to the least fixpoint of $F_{\mathtt{DMS}}$.

Proof. Let $I^\sharp = (I_{v_1}, \ldots, I_{v_m}), Msg)$ be any distributed interpretation. Suppose $F_{\mathtt{DMS}_0}(I^\sharp) = ((J_{v_1}, \ldots, J_{v_m}), Msg_1)$ and $F_{\mathtt{DMS}}(I^\sharp) = ((H_{v_1}, \ldots, H_{v_m}), Msg_2)$. It follows immediately from Theorem 3 that for all distributed interpretations $I^\sharp$, $F_{\mathtt{DMS}}(I^\sharp) \leq F_{\mathtt{DMS}_0}(I^\sharp)$ because for all $C \in \mathbf{M}$, $md'_{\mathbf{N}}(C) \subseteq md_{\mathbf{N}}(C)$. Consequently, it follows immediately that $\mathsf{lfp}(F_{\mathtt{DMS}}) \leq \mathsf{lfp}(F_{\mathtt{DMS}_0})$. Hence, we only need to show that $\mathsf{lfp}(F_{\mathtt{DMS}_0}) \leq \mathsf{lfp}(F_{\mathtt{DMS}})$.

We prove, by induction, that for all ordinals γ, $F_{\mathtt{DMS}_0} \uparrow \gamma \leq \mathsf{lfp}(F_{\mathtt{DMS}})$. We use $((J^\gamma_{v_1}, \ldots, J^\gamma_{v_m}), Msg^\gamma_1)$ to denote $F_{\mathtt{DMS}_0} \uparrow \gamma$ and $((H^\gamma_{v_1}, \ldots, H^\gamma_{v_m}), Msg^\gamma_2)$ to denote $F_{\mathtt{DMS}} \uparrow \gamma$. We use $H_{v_i}^{\mathsf{lfp}}$ (resp. $J v_i{}^{\mathsf{lfp}}$ to denote the value of $H^\gamma_{v_i}$ (resp. $J^\gamma_{v_i}$) where γ is the closure ordinal of $F_{\mathtt{DMS}_0}$ (resp. $F_{\mathtt{DMS}}$).

$\boxed{\text{Base Case } (\gamma = 0)}$ Immediate.

$\boxed{\text{Inductive Case}}$ There are two subcases, when $\gamma = (\eta + 1)$ is a successor ordinal, and when γ is a limit ordinal.

Subcase 1 ($\gamma = (\eta + 1)$ is a successor ordinal): Suppose $J^\gamma_{v_i}(A)(D)(s) = \mu$ for some $1 \leq i \leq m$. Then $J^\gamma_{v_i} = F_{\mathtt{DMS}_0}(J^{\eta,\sharp}_{v_i})$ where $J^{\eta,\sharp}_{v_i} = T_{Q'_{v_i}}(J^\eta_{v_i})$ where Q'_{v_i} is the amalgamated site knowledge base constructed in part (2) of Definition 13 w.r.t. the messages in Msg^η_1. By the induction hypothesis, $J^\eta_{v_i} \leq H^{\mathsf{lfp}}_{v_i}$. Hence, by the monotonicity of $T_{Q_{v_i}}$, it follows that

$$J^{\eta,\sharp}_{v_i} = T_{Q'_{v_i}}(J^\eta_{v_i}) \leq T_{Q'_{v_i}}(H^{\mathsf{lfp}}_{v_i}) = H^{\mathsf{lfp}}_{v_i}.$$

If we can show that $Q'_{v_i} \subseteq Q''_{v_i}$ where Q''_{v_i} is the amalgamated knowledge base constructed in part (2) of Definition 13 w.r.t. the messages in Msg^η_1, then we would be done because

$$T_{Q'_{v_i}}(H^\eta_{v_i}) \leq T_{Q''_{v_i}}(H v_i{}^\eta) = H^\gamma_{v_i}.$$

Suppose $C \in Q'_{v_i}$. Then either $C \in Q_{v_i}$ where Q_{v_i} is the amalgamation of the clauses in v_i, in which case $C \in Q''_{v_i}$ follows immediately, or C is a unit clause of the form $A : [D, \mu, t]$ such that $Tell^{v_j, v_i}(A : [D, \mu, t]) \in Msg^\eta_1$ for some site v_j. As $Msg^\eta_1 \subseteq Msg^\eta_2$ by the induction hypothesis, it follows that $Tell^{v_j, v_i}(A : [D, \mu, t]) \in Msg^\eta_2$ and hence, $A : [D, \mu, t] \in Q''_{v_i}$.

It remains to show that $Msg^\gamma_1 \subseteq Msg^\gamma_2$. Suppose $Tell^{v_i, v_j}(A : [D, \mu, t]) \in Msg^\gamma_1$. If $Tell^{v_i, v_j}(A : [D, \mu, t]) \in Msg^\eta_1$, then we are done immediately by the induction hypothesis. Otherwise, $A : [D, \mu, t]$ is M-satisfied by $T_{Q'_{v_i}}(J_{v_i})$ and hence, there is a set of clauses in Q'_{v_i} such that the bodies of these clauses are true in $G_{v_i}(J_{v_i})$ and the heads of these clauses jointly imply $A : [D, \mu, t]$. By the induction hypothesis, these clause bodies are true in H_{v_i}. Suppose $B_1 : [D_1, \mu_1, t_1]$ is an annotated atom in the body of one of these clauses. By the definition of acceptable placement, every clause in $DB_1, \ldots, DB_n$ whose head $B_2 : [D_2, \mu_2, t_2]$ is such that B_1 and B_2 are unifiable and $D_2 \subseteq D_1$ is in the access set of v_i. Hence, all these clauses must be present in v_z for some $v_z \in \mathtt{ACCESS}(v_i, B_1 : [D_1, \mu_1, t_1])$. Hence, $Tell^{v_z, v_j}(B_1 : [D_1, \mu_1, t_1])$ is true in Msg^γ_2 and hence, $Tell^{v_i, v_j}(B_1 : [D_1, \mu_1, t_1])$ is true in $Msg^{\gamma+\ell}_2$ where ℓ is the length of the path from v_z to v_j, i.e. $Tell^{v_i, v_j}(B_1 : [D_1, \mu_1, t_1]) \in Msg_2$, and we are done.

Subcase 2 (γ is a limit ordinal): $J^\gamma_{v_i}$ is the lub of $J^\beta v_i$ for $\beta < \gamma$ and as $J^\beta v_i \leq H^\beta v_i$ by the induction hypothesis, the result follows immediately. $\square$

Note that Theorem 4 holds even if we consider the case when $md'_{\mathbf{N}}(C)$ is a semi-placement for all clauses C.

It is important to note that there may be, in general, several sets of sites which satisfy the conditions for it (i.e. the set of sites) to be an acceptable placement for clause C. The following example illustrates how this may happen.

Example 9 Recall the robot example, $\mathtt{DMS}_R$, given in the appendix. The following mediating clause was located at site 3:

$$\begin{aligned}
&temperature(X,Y)\colon [\{\mathbf{m}\}, V_1 \sqcup V_2, \{V_t\}] \leftarrow \\
&\qquad temperature(X,Y) : [\{3\}, V_1, \{V_{t_1}\}]\ \& \\
&\qquad temperature(X,Y) : [\{4\}, V_2, \{V_{t_2}\}].
\end{aligned}$$

Since the only clauses that unify with the atoms in the body of this clause are located at site v_3 and as v_3 is in the $\mathtt{ACCESS}_{\mathtt{DMS}_R}$ set of both v_1 and v_2 for these atoms, $\{v_1\}$,$\{v_2\}$ and $\{v_3\}$ are all acceptable placements for this clause.

Let us now consider the following clause which is also located at site v_3:

$$\begin{aligned}
&temperature(X,Y_1)\colon [\{\mathbf{m}\}, V_1, \{V_t\}] \leftarrow \\
&\qquad Y_1 > Y_2 \,\| \\
&\qquad recent_temperature(X,Y_1) : [\{\mathbf{m}\}, V_1, \{V_t\}]\ \& \\
&\qquad temperature(X,Y_2) : [\{5\}, V_2, \{V_{t_2}\}].
\end{aligned}$$

Suppose the clauses containing information about the predicate *recent_temperature* are stored at site 3 as given. Then, $\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_3, recent\_temperature(X,Y_1) : [\{\mathbf{m}\}, V_1, \{V_t\}]) = \{v_3\}$ and both $\{v_2\}$ and $\{v_3\}$ are acceptable placements for the above clause since $\{v_3\}$ is contained in the ACCESS set of both v_2 and v_3 for all the atoms in the body of the clause. $\{v_1\}$ however is not an acceptable placement, since $\mathtt{ACCESS}_{\mathtt{DMS}_R}(v_1, recent\_temperature(X,Y_1) : [\{\mathbf{m}\}, V_1, \{V_t\}]) = \{v_1\}$ and $\{v_3\} \not\subseteq \{v_1\}$. □

6.2. Distributing Mediators when Link Failures may Occur

In this section, we consider the case when we have a distributed mediated system

$$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_m\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

and $i \geq 0$ links are allowed to fail. Intuitively, when a link between sites v_1 and v_2 fails, this means that E above is modified to $(E - \{(v_1, v_2)\})$. A system designer may reason thus: "Suppose, in my worst dreams, at most i links in the network can fail. I would like to distribute the mediator in such a way (if possible) that even if i links go down, the system functions appropriately. Are there ways of identifying the circumstances under which this is possible ?" This is the question addressed in this section.

Definition 17 Suppose

$$\mathtt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_m\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is a distributed mediated system, i is an integer, and for all $C \in \mathbf{M}$, $md_{\mathbf{N}}$ is an acceptable placement for C. A mediatory distribution function $md'_{\mathbf{N}}$ is said to be *resilient w.r.t.* DMS *for at most i link failures* iff:

(i) $md'_{\mathbf{N}}$ is an acceptable placement of C w.r.t. $\mathtt{DMS}_Y$ for all subsets $Y \subseteq E$ of cardinality $card(E) - i$ and (w.r.t. $\mathtt{DMS}$) for all $C \in \mathbf{M}$, and

(ii) for all subsets $X \subseteq E$ that are of cardinality i, the distributed mediated system $\mathtt{DMS}_X$ which is identical to $\mathtt{DMS}$ except that E is replaced by $(E - X)$ and $md_{\mathbf{N}}$ with $md'_{\mathbf{N}}$ has the property that the least fixpoint of $F_{\mathtt{DMS}_X}$ coincides with the least fixpoint of $F_{\mathtt{DMS}}$.

Intuitively, the integer i specifies an upper bound on how many links are assumed to go down (in the worst-case). The links in X are the edges that are assumed to go down. When the links in X go down, the network effectively consists of the edges in $(E - X)$. A distribution, $md'_{\mathbf{N}}$, of mediating clauses achieves the same effect as the original distributed mediated system iff the least fixpoint of the operator associated with the original system (i.e. $F_{\mathtt{DMS}}$) coincides with the least fixpoint of the operator associated with the system (whose links are down). As the identity of the links that go down cannot be predicted in advance, all possible collections of i links in E need to be considered.

Example 10 Suppose $\mathtt{DMS}$ is a distributed mediated system that has access to four deductive databases, DB_1 only contains information about the atom $p(X)$, DB_2 about $q(X)$, DB_3 about $r(X)$ and finally DB_4 about $s(X)$. Suppose the system has three sites: v_1 has access to both DB_1 and DB_2,v_2 to both DB_2 and DB_3 and v_3 to DB_3 and DB_4. Sites are connected in a ring structure, i.e. E contains the following edges: $\{(v_1, v_2), (v_2, v_3), (v_3, v_1), (v_1, v_3), (v_3, v_2), (v_2, v_1)\}$.

Suppose f_{Ask} is given as follows:

$$\begin{aligned} f_{Ask}(v_1, q(X) : [\{2\}, V, T]) &= \{v_2, v_3\} \\ f_{Ask}(v_2, p(X) : [\{1\}, V, T]) &= \{v_1\} \\ f_{Ask}(v_2, r(X) : [\{3\}, V, T]) &= \{v_3\} \\ f_{Ask}(v_3, q(X) : [\{2\}, V, T]) &= \{v_1, v_2\} \end{aligned}$$

We want to distribute the following mediating clauses such that the resulting system is resilient w.r.t. $\mathtt{DMS}$ for at most 1 link failure. A suitable mediatory distribution function $md_{\mathbf{N}}$ can be given as follows: Place the first clause at site v_1, second clause at site v_2 and the third clause at site v_3.

$$\begin{aligned} d_1(X) : [\{\mathbf{m}\}, V, T] &\leftarrow p(X) : [\{1\}, V, T] \,\&\, q(X) : [\{2\}, V, T] && (1) \\ d_2(X) : [\{\mathbf{m}\}, V, T] &\leftarrow q(X) : [\{2\}, V, T] \,\&\, r(X) : [\{3\}, V, T] && (2) \\ d_3(X) : [\{\mathbf{m}\}, V, T] &\leftarrow q(X) : [\{2\}, V, T] \,\&\, r(X) : [\{3\}, V, T] && \\ & s(X) : [\{4\}, V, T] && (3) \end{aligned}$$

Note that although site v_2 is also an acceptable placement for the first mediating clause, the system resulting from placing this clause only at this site is not even resilient to 1 link failure. □

The following result states that if $md'_{\mathbf{N}}$ is an acceptable placement for a system with "down" links, then $md'_{\mathbf{N}}$ must have been a semi-placement of the original system. This means that acceptable placements for a system with "down" links must be selected from the semi-placements of the original system.

Theorem 5 Suppose

$$\mathtt{DMS}_{\emptyset} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_m\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is a distributed mediated system, and suppose $i \geq 0$ links are allowed to go "down". Given a clause C, if $md'_{\mathbf{N}}$ is an acceptable placement for C w.r.t. $\mathtt{DMS}_X$ (as specified in Definition 17) for some $X \subseteq E$ such that $card(X) = card(E) - i$ and f_{Ask} preserves subsumption, then $md'_{\mathbf{N}}$ is a semi-placement for C w.r.t. $\mathtt{DMS}_{\emptyset}$.

Proof: We will prove this by showing that $md'_{\mathbf{N}}$ satisfies conditions (1) and (2) of definition 16. Proving conditions (1) and (2) are symmetric except that condition (2) holds even if f_{Ask} is arbitrary. We will first construct a graph $A_{\mathtt{DMS}}(A : [D, \mu, t])$ given an arbitrary distributed system $\mathtt{DMS}$ and show how the $\mathtt{ACCESS}$ set relates to this graph. Then, we will prove that $md'_{\mathbf{N}}$ satisfies the first condition in definition 16 for clause C w.r.t. $\mathtt{DMS}_{\emptyset}$.

Suppose $\mathtt{DMS}$ is a distributed mediated system. We construct the directed graph $A_{\mathtt{DMS}}(A : [D, \mu, t]) = (V_{\mathtt{DMS}}, E_{\mathtt{DMS}})$ as follows: Set $V_{\mathtt{DMS}} = V$ and $E_{\mathtt{DMS}} = \emptyset$, do the following for all the edges $(v_i, v_j) \in E$: if $v_j \in f_{Ask}(v_i, A : [D, \mu, t])$, then add the directed edge (v_i, v_j) (from v_i to v_j) to $E_{\mathtt{DMS}}$. Since the network of $\mathtt{DMS}$ is undirected, E contains symmetric pairs, hence v_j will be considered in the pair (v_j, v_i).

Given the graph $A_{\mathtt{DMS}}(A : [D, \mu, t]) = (V_{\mathtt{DMS}}, E_{\mathtt{DMS}})$ constructed as above, let $R_{\mathtt{DMS}}(v_i, A : [D, \mu, t]) = \{v_j \mid v_j$ is reachable from v_i via a path in $A_{\mathtt{DMS}}(A : [D, \mu, t])\}$, then $\mathtt{ACCESS}_{\mathtt{DMS}}(v_i, A : [D, \mu, t]) = R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$.

We prove this using the definition of $\mathtt{ACCESS}$ given in definition 15.

($\Rightarrow$) If $v \in \mathtt{ACCESS}_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$ then $v \in R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$:

(i) Since every node v_i is reachable from itself, $v_i \in R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$ for all $v_i \in V$.

(ii) If $v_j \in f_{Ask}(v_i, A : [D, \mu, t])$ and $(v_i, v_j) \in E$ then there is an edge $(v_i, v_j) \in E_{\mathtt{DMS}}$ and $v_j \in R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$. To show that $R_{\mathtt{DMS}}(v_j, A : [D, \mu, t]) \subseteq R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$ we consider two cases:

- $v_i \in R_{\mathtt{DMS}}(v_j, A : [D, \mu, t])$, then it must be the case that there is a path from v_j to v_i. We know that there is a path from v_i to v_j as well via link (v_i, v_j). Then, v_i and v_j reach exactly the same sites, therefore $R_{\mathtt{DMS}}(v_j, A : [D, \mu, t]) = R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$.
- $v_i \notin R_{\mathtt{DMS}}(v_j, A : [D, \mu, t])$, then we know that $\{v_i\} \cup R_{\mathtt{DMS}}(v_j, A : [D, \mu, t]) \subseteq R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$, and $R_{\mathtt{DMS}}(v_j, A : [D, \mu, t]) \subseteq R_{\mathtt{DMS}}(v_i, A : [D, \mu, t])$.

($\Leftarrow$) If $v \in R_{\texttt{DMS}}(v_i, A : [D, \mu, t])$ then $v \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$: In this case, there is a path from v_i to v in $A_{\texttt{DMS}}(A : [D, \mu, t])$. For all the edges $(v_{a_i}, v_{a_{i+1}})$ on this path we have that $v_{a_{i+1}} \in f_{Ask}(v_{a_i}, A : [D, \mu, t])$, hence if $v_{a_i} \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$ then it must be the case that $v_{a_{i+1}} \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$. From the definition of the ACCESS set, $v_i \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$ and consequently $v \in \texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t])$. Hence, we have proven that $\texttt{ACCESS}_{\texttt{DMS}}(v_i, A : [D, \mu, t]) = R_{\texttt{DMS}}(v_i, A : [D, \mu, t])$.

Given a mediating clause C, let $A : [D, \mu, t]$ be an atom in the body of C and $ASKB(v_j)$ contain a clause having the head $A^* : [D^*, \mu^*, t^*]$ such that A and A^* are unifiable via mgu θ and $D^* \subseteq D$. Since $md'_{\mathbf{N}}$ is an acceptable placement for C w.r.t. $\texttt{DMS}_X$, we have that:

$$\texttt{ACCESS}_{\texttt{DMS}_X}(v_j, A^* : [D^*, \mu^*, t^*]\theta) \subseteq \bigcup_{v \in md'_{\mathbf{N}}} \texttt{ACCESS}_{\texttt{DMS}_X}(v, A : [D, \mu, t]).$$

This means that all sites in $\texttt{ACCESS}_{\texttt{DMS}_X}(v_j, A^* : [D^*, \mu^*, t^*]\theta)$ are reachable from a site in X in $A_{\texttt{DMS}_X}(A : [D, \mu, t]) = (E_{\texttt{DMS}_X}, V_{\texttt{DMS}_X})$. We want to prove that

$$\texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v_j, A^* : [D^*, \mu^*, t^*]\theta) \subseteq \bigcup_{v \in md'_{\mathbf{N}}} \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v, A : [D, \mu, t]).$$

Since $\texttt{DMS}_\emptyset$ contains i more links than $\texttt{DMS}_X$, we have that

$$R_{\texttt{DMS}_X}(v_j, A^* : [D^*, \mu^*, t^*]\theta) \subseteq R_{\texttt{DMS}_\emptyset}(v_j, A^* : [D^*, \mu^*, t^*]\theta).$$

Therefore, $\texttt{ACCESS}_{\texttt{DMS}_X}(v_j, A^* : [D^*, \mu^*, t^*]\theta) \subseteq \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v_j, A^* : [D^*, \mu^*, t^*]\theta)$. Similarly, for all $v \in V$ we have $\texttt{ACCESS}_{\texttt{DMS}_X}(v, A : [D, \mu, t]) \subseteq \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v, A : [D, \mu, t])$.

Assume by way of contradiction that $v \in \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v_j, A^* : [D^*, \mu^*, t^*]\theta)$ and $v \notin \bigcup_{v \in md'_{\mathbf{N}}} \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v, A : [D, \mu, t])$. Then, it must be the case that v is not reachable from any site in X in $A_{\texttt{DMS}_\emptyset}(A : [D, \mu, t])$. Since $v \in \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v_j, A^* : [D^*, \mu^*, t^*]\theta)$, it is reachable from a site $v' \in \texttt{ACCESS}_{\texttt{DMS}_X}(v_j, A^* : [D^*, \mu^*, t^*]\theta)$. Since $\bigcup_{v \in md'_{\mathbf{N}}} \texttt{ACCESS}_{\texttt{DMS}_X}(v, A : [D, \mu, t]) \subseteq \bigcup_{v \in md'_{\mathbf{N}}} \texttt{ACCESS}_{\texttt{DMS}_\emptyset}(v, A : [D, \mu, t])$, all such v' is reachable from some site $v_X \in X$ in $A_{\texttt{DMS}_\emptyset}(A : [D, \mu, t]))$. Let $v_{l_1} = v'$ and $v_{l_d} = v$ and $v_{l_1}, \ldots, v_{l_d}$ be a path from v' to v in $A_{\texttt{DMS}_\emptyset}(A^* : [D^*, \mu^*, t^*]\theta)$. Then, for all the edges $(v_{l_i}, v_{l_{i+1}}) \in E$ on this path, we have that $v_{l_{i+1}} \in f_{Ask}(v_{l_i}, A^* : [D^*, \mu^*, t^*]\theta)$. Since $A : [D, \mu, t]$ subsumes $A^* : [D^*, \mu^*, t^*]\theta$, and f_{Ask} preserves subsumption, it must be the case that $v_{l_{i+1}} \in f_{Ask}(v_{l_i}, A : [D, \mu, t])$, and $v_{l_{i+1}}$ is reachable from v_{l_i} in $A_{\texttt{DMS}_\emptyset}(A : [D, \mu, t])$. Hence, v is reachable from v' in $A_{\texttt{DMS}_\emptyset}(A : [D, \mu, t])$ and consequently v is reachable from site $v_X \in X$ in $A_{\texttt{DMS}_\emptyset}(A : [D, \mu, t])$ which contradicts with the statement above. $\Box$

The following result is an immediate consequence of Theorem 5 and Theorem 4.

Corollary 3 Suppose

$$\texttt{DMS} = (V, E, \{DB_1, \ldots, DB_n, \Sigma_1, \ldots, \Sigma_m\}, f_{\mathbf{N}}, f_{Ask}, md_{\mathbf{N}}, \mathbf{M})$$

is a distributed mediated system where f_{Ask} preserves subsumption, and i is an integer. Then there exists a mediatory distribution function that is resilient w.r.t. DMS for at most i link failures iff there exists a semi-placement $md'_{\mathbf{N}}$ w.r.t. DMS which is resilient for 0 link failures w.r.t. $\texttt{DMS}_X$ for all $X \subseteq E$ of cardinality $card(E) - i$. □

The above corollary says, in effect, that looking at the semi-placements of clauses in the mediator **M** is sufficient to determine whether it is possible to guard against i link failures. An algorithm to perform such a check can be immediately devised in the following way:

(i) Find a semi-placement, $md'_{\mathbf{N}}$, of DMS that is different from previously generated semi-placements of DMS. If no such new semi-placements exist, then **halt** and return **fail**.

(ii) If, for all $X \subseteq E$ of cardinality i, it is the case that $md'_{\mathbf{N}}$ is an acceptable placement for $\texttt{DMS}_X$, then **halt with success**, and return $md'_{\mathbf{N}}$.

(iii) Otherwise return to Step 1.

The above skeletal algorithm can be "fine-tuned" in many ways. However, in the worst case, the problem of computing a resilient mediatory distribution function, may be exponential in the number of links in the network as there are, in general $\binom{n}{i}$ ways in which i links may go down (where n is the number of links in E). Fortunately, this algorithm needs to be executed only once, when the mediator is being distributed (though incremental modifications may need to be performed when new nodes and/or databases are added to DMS).

7. Related Work

The idea of mediators and distributed mediators is due to Gio Wiederhold [26, 27] who proposed that a program, called a mediator, should be used to inter-operate between multiple representations of knowledge and data, both in distributed, as well as in centralized environments.

A great deal of work has been done in *multidatabase* systems and *interoperable* database systems[10, 24, 29]. However, most of this work combines standard relational databases (no deductive capabilities). Not much has been done on the development of a semantic foundation for such databases. The work of Grant et. al. [10] is an exception: the authors develop a calculus and an algebra for integrating information from multiple databases. This calculus extends the standard relational calculus. Further work specialized to handle inter-operability of multi-databases is critically needed. However, our paper addresses a different topic – that of integrating multiple deductive databases containing (possibly) inconsistencies, uncertainty, non-monotonic negation, and possibly even temporal information. Zicari et. al [29] describe how interoperability may be achieved between a rule-based system (deductive DB) and an object-oriented database using special *import/export* primitives. No formal theory is developed in [29]. Perhaps closer to our goal is that of Whang et. al. [24] who argue that Prolog is a suitable framework for schema integration. In fact, the approach of Whang et. al. is in the same spirit as that of metalogic programming discussed earlier. Whang et. al. do not give a formal semantics for multi-databases containing inconsistency and/or uncertainty and/or non-monotonicity and/or temporal information. Reasoning with temporal mismatches has been studied by Jajodia and Wiederhold and their colleagues [28, 25]. This work complements ours and it would be interesting, in future work, to see how these ideas can be expressed in our framework.

Dubois, Lang and Prade [7], also suggest that formulas in knowledge bases can be annotated with, for each source, a lower bound of a degree of certainty associated with that source. The spirit behind their approach is similar to ours, though interest is restricted to the $[0, 1]$ lattice, the stable and well-founded semantics are not addressed, and amalgamation theorems are not studied. However, for the $[0, 1]$ case, their framework is a bit richer than ours when nonmonotonic negations are absent. The authors have extended their work to accommodate time in [6].

Previous Work of Authors: This paper forms part of a long-term project on developing a formal theoretical foundation, as well as algorithms, implementations, and applications of mediated information systems. In [23], Subrahmanian proposed a formal logical framework for integrating multiple knowledge bases, and showed that this framework could be used to represent and manipulate certain forms of time and uncertainty, as well as nonmonotonicity. Subsequently, Nerode and Subrahmanian [17] proposed the notion of a hybrid knowledge base where inter-operating with auxiliary data structures and constraint domains was also accomplished. Adalı and Subrahmanian developed this logical framework further, giving a set of data structures and algorithms that could be interrupted in the middle of a computation (if necessary). Such an interrupt would cause an intermediate, approximate answer to be returned. This paper uses the same logical framework described in the above works, with one major difference. In reality, databases are likely to be located at different sites on a network (such as a LAN or the Internet). Hence, though the mediator-based framework in [23, 17] specifies the *declarative* result of such a logic computation, it does not specify a distributed realization of this declarative semantics. This is what has been accomplished in this paper. In addition, this paper also addresses conditions under which this distributed semantics is robust, i.e. continues to behave appropriately even if some links in the network "go down."

8. Conclusions

In [23, 17], we provided a formal declarative semantics for integrating multiple databases. Concurrently with this paper, [1], provides a formal operational procedure that is interruptable (and will give approximate answers when interrupted) and that caches previous computations so as to eliminate redundant computations. This formal theory is now leading to an application for missile siting by the US Army Corps of Engineers[3].

In this paper, we have extended the theory of mediators developed in [23, 17, 1, 3] to the case when the databases being mediated between are stored at different nodes in a network (such as a LAN or the Internet). We have developed a distributed semantics for such mediated databases, and shown when such a distributed semantics is equivalent to the non-distributed semantics. Declaratively, such an equivalence result is of extreme importance because the physical location of the databases in the mediated system should be irrelevant as far as the quality of answers provided to the user is concerned. The user expects the right answer to his/her query, independently of where along the network a particular database is located.

Subsequently, we show conditions under which the above distributed semantics is equivalent to the nondistributed semantics, even when a certain (pre-specified) number of links in the network are allowed to "go down." What this means is that if the system designer believes that in the worst case, i links in the network may go down, then s/he may choose to use our notion of a semi-placement to distribute the mediator (under the conditions specified in the paper).

References

[1] S. Adalı and V.S. Subrahmanian. (1993) *Amalgamating Knowledge Bases, III: Algorithms, Data Structures and Query Processing*, Univ. of Maryland CS-TR-3124, Aug. 1993. Submitted for journal publication.

[2] R. Agrawal, R. Cochrane and B. Lindsay. (1991) *On Maintaining Priorities in a Production Rule System*, Proc. VLDB-91, pps 479–487.

[3] J. Benton and V.S. Subrahmanian. (1993) *Hybrid Knowledge Bases for Missile Siting Problems*, accepted for publication in 1994 Intl. Conf. on Artificial Intelligence Applications, IEEE Press.

[4] Y. Breitbart, H. Garcia-Molina and A. Silberschatz. (1992) *Overview of Multidatabase Transaction Management*, VLDB Journal, 2, pps 181–239.

[5] D. Dubois, J. Lang and H. Prade. (1991) *Towards Possibilistic Logic Programming*, Proc. 1991 Intl. Conf. on Logic Programming, ed. K. Furukawa, pps 581–595, MIT Press.

[6] D. Dubois, J. Lang and H. Prade. (1991) *Timed Possibilistic Logic*, Fundamenta Informaticae, Vol. XV, No. 3-4, pps 211-234.

[7] D. Dubois, J. Lang and H. Prade. (1992) *Dealing with Multi-Source Information in Possibilistic Logic*, Proc. 10th European Conf. on Artificial Intelligence, Wiley.

[8] R. Fagin, J.D. Ullman, and M.Y. Vardi. (1983) *On the Semantics of Updates in Databases*, Proc. ACM SIGACT/SIGMOD Symposium on Principles of Database Systems, pps 352–365.

[9] R. Fagin, G. Kuper, J. Ullman, and M. Vardi. (1986) *Updating Logical Databases*, In *Advances in Computing Research*, volume 3, pages 1–18, 1986.

[10] J. Grant, W. Litwin, N. Roussopoulos and T. Sellis. (1991) *An Algebra and Calculus for Relational Multidatabase Systems*, Proc. First International

Workshop on Interoperability in Multidatabase Systems, IEEE Computer Society Press (1991) 118-124.

[11] R. Hahnle. (1990) *Towards an Efficient Tableau Rules for Multiple-Valued Logics*, Proc. Workshop on Computer Science Logic, pps 248–260, Springer.

[12] R. Hahnle. (1990) *Uniform Notation Tableau Rules for Multiple-Valued Logics*, Proc. 1991 IEEE Intl. Symp. on Multiple Valued Logic, pps 26–29.

[13] W. Kim and J. Seo. (1991) *Classifying Schematic and Data Heterogeneity in Multidatabase Systems*, IEEE Computer, Dec. 1991.

[14] M. Kifer and V.S. Subrahmanian. (1989) *Theory of Generalized Annotated Logic Programming and its Applications*, Journal of Logic Programming, 12, 4, pps 335–368, 1992. Preliminary version in: Proc. 1989 North American Conf. on Logic Programming, MIT Press.

[15] R. Krishnamurthy, W. Litwin and W. Kent. (1991) *Language Features for Interoperability of Databases with Schematic Discrepancies*, Proc. ACM SIGMOD 1991.

[16] A. Lefebvre, P. Bernus and R. Topor. (1992) *Querying Heterogeneous Databases: A Case Study*, draft manuscript.

[17] J. Lu, A. Nerode, and V.S. Subrahmanian. (1993) *Hybrid Knowledge Bases*, Univ. of Maryland CS-TR-3037. Submitted for journal publication.

[18] Z. Manna. *Mathematical Theory of Computation*, Prentice-Hall.

[19] N. Murray and E. Rosenthal. (1993) *Signed Formulas: A Liftable Meta-Logic for Multiple Valued Logics*, Proc. ISMIS-1993, Springer Verlag.

[20] H. Samet. (1989) *The Design and Analysis of Spatial Data Structures*, Addison-Wesley.

[21] A. Sheth and J. Larson. (1990) *Federated Database Systems for Managing Distributed, Heterogeneous and Autonomous Databases*, ACM Computing Surveys, 22, 3, pp 183–236.

[22] A. Silberschatz, M. Stonebraker and J. D. Ullman. (1991) *Database Systems: Achievements and Opportunities*, Comm. of the ACM, 34, 10, pps 110–120.

[23] V.S. Subrahmanian. (1994) *Amalgamating Knowledge Bases*, ACM Trans. on Database Systems, 19, 2, pps 291–331.

[24] W.K. Whang, S. B. Navathe and S. Chakravarthy. (1991) *Logic-Based Approach for Realizing a Federated Information System*, Proc. First International Workshop on Interoperability in Multidatabase Systems, IEEE Computer Society Press (1991) 92–100.

[25] G. Wiederhold, S. Jajodia, and W. Litwin. Dealing with granularity of time in temporal databases. In *Proc. 3rd Nordic Conf. on Advanced Information Systems Engineering*, Lecture Notes in Computer Science, Vol. 498, (R. Anderson et al. eds.), Springer-Verlag, 1991, pages 124–140.

[26] G. Wiederhold. (1992) *Mediators in the Architecture of Future Information Systems*, IEEE Computer, March 1992, pps 38–49.

[27] G. Wiederhold. (1993) *Intelligent Integration of Information*, Proc. 1993 ACM SIGMOD Conf. on Management of Data, pps 434–437.

[28] G Wiederhold, S. Jajodia, and W. Litwin. Integrating temporal data in a heterogeneous environment. In *Temporal Databases.* Benjamin/Cummings, Jan 1993.

[29] R. Zicari, S. Ceri, and L. Tanca. (1991) *Interoperability between a Rule-Based Database Language and an Object-Oriented Language*, Proc. First International Workshop on Interoperability in Multidatabase Systems, IEEE Computer Society Press (1991) 125-135.

Appendix: The complete robot example

In this section, we give the complete list of clauses stored in different sites. Note that the mediating clauses stored in Site 1 access the spatial data structure when evaluating the `RANGE` subquery and the at and in relations. Similarly, at Site 2, the relational database is used to process the *weight* and *width* relations and finally the real number constraint domain is used to evaluate the constraints involving numeric expressions.

Site 1:

$$
\begin{aligned}
&cool\_spray(Obj, X, Y)\colon [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad (X_1, Y_1)\,\underline{\text{in}}\,\mathtt{RANGE}((\mathtt{X},\mathtt{Y}), 2) \,\&\, \underline{\text{at}}(Obj, X_1, Y_1) \,\&\, \underline{\text{at}}(r1, X, Y) \,\|\\
&\qquad temperature(Obj, T) : [\{\mathbf{m}\}, 0.5, \{V_t\}] \,\&\, T \geq 90\ .\\
&max\_possible\_speed(R, Obj, S) : [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad \underline{\text{at}}(R, X, Y) \,\&\, \underline{\text{at}}(Obj, X_1, Y_1) \,\&\\
&\qquad S = |X_1 - X| * S_2 + |Y_1 - Y| * S_1 \,\|\\
&\qquad max\_speed(R, vertical, S_1) : [\{1, 2\}, 1, \mathbf{R}^+] \,\&\\
&\qquad max\_speed(R, horizontal, S_2) : [\{1, 2\}, 1, \mathbf{R}^+]\ .\\
&command\_move(R, Obj) : [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad can\_lift(R, Obj)[\{\mathbf{m}\}, 1, \{V_t\}] \,\&\\
&\qquad can\_lift(R_1, Obj)[\{\mathbf{m}\}, 1, \{V_t\}] \,\&\, R_1 \neq R \,\&\\
&\qquad max\_possible\_speed(R, Obj, S) : [\{\mathbf{m}\}, 1, \{V_t\}] \,\&\\
&\qquad max\_possible\_speed(R_1, Obj, S_1) : [\{\mathbf{m}\}, 1, \{V_t\}] \,\&\, S \geq S_1.\\
&command\_move(R, Obj) : [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad can\_lift(R, Obj)[\{\mathbf{m}\}, 1, \{V_t\}] \,\&\, R_1 \neq R \,\&\\
&\qquad \mathbf{not}(can\_lift(R_1, Obj)[\{\mathbf{m}\}, 1, \{V_t\}]).
\end{aligned}
$$

Site 2:

$$
\begin{aligned}
&can\_lift(r1, Obj) \colon [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad weight(Obj, W_1) \,\&\, W \geq W_1 \,\&\, width(Obj, D_1) \,\&\, D \geq D_1 \,\| \\
&\qquad max\_weight\_capability(r1, W) : [\{1\}, 1, \mathbf{R}^+] \,\& \\
&\qquad max\_distance\_between\_arms(r1, D) : [\{1\}, 1, \mathbf{R}^+] \,\& \\
&\qquad max\_temperature\_handling(r1, T) : [\{1\}, 1, \mathbf{R}^+] \,\& \\
&\qquad temperature(Obj, T_1) : [\{\mathbf{m}\}, 0.9, \{V_t\}] \,\&\, T_1 \leq T. \\
&can\_lift(r2, Obj) : [\{\mathbf{m}\}, 1, \{V_t\}] \leftarrow \\
&\qquad weight(Obj, W_1) \,\&\, W \geq W_1 \,\&\, width(Obj, D_1) \,\&\, D \geq D_1 \,\| \\
&\qquad max\_weight\_capability(r2, W) : [\{2\}, 1, \mathbf{R}^+] \,\& \\
&\qquad max\_distance\_between\_arms(r2, D) : [\{2\}, 1, \mathbf{R}^+] \,\& \\
&\qquad max\_temperature\_handling(r2, T) : [\{2\}, 1, \mathbf{R}^+] \,\& \\
&\qquad temperature(Obj, T_1) : [\{\mathbf{m}\}, 0.8, \{V_t\}] \,\&\, T_1 \leq T. \\
&at(r1, 1, 2) : [\{1\}, 1, \{0\}] \leftarrow . \\
&max\_weight\_capability(r1, 100) : [\{1\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_temperature\_handling(r1, 65) : [\{1\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_distance\_between\_arms(r1, 20) : [\{1\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_speed(r1, vertical, 1) : [\{1\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_speed(r1, horizontal, 2) : [\{1\}, 1, \mathbf{R}^+] \leftarrow . \\
&at(r2, 3, 1) : [\{2\}, 1, \{0\}] \leftarrow . \\
&max\_weight\_capability(r2, 50) : [\{2\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_temperature\_handling(r2, 90) : [\{2\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_distance\_between\_arms(r2, 50) : [\{2\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_speed(r2, vertical, 2) : [\{2\}, 1, \mathbf{R}^+] \leftarrow . \\
&max\_speed(r2, horizontal, 0.5) : [\{2\}, 1, \mathbf{R}^+] \leftarrow .
\end{aligned}
$$

Site 3:

$$
\begin{aligned}
&recent\_temperature(X, Y_1) \colon [\{\mathbf{m}\}, V_1, \{V_{t_1}\}] \leftarrow \\
&\qquad temperature(X, Y_1) : [\{3\}, V_1, \{V_{t_1}\}] \,\& \\
&\qquad temperature(X, Y_2) : [\{4\}, V_2, \{V_{t_2}\}] \,\&\, V_{t_1} > V_{t_2}. \\
&recent\_temperature(X, Y_2) : [\{\mathbf{m}\}, V_2, \{V_{t_2}\}] \leftarrow \\
&\qquad temperature(X, Y_1) : [\{3\}, V_1, \{V_{t_1}\}] \,\& \\
&\qquad temperature(X, Y_2) : [\{4\}, V_2, \{V_{t_2}\}] \,\&\, V_{t_2} \geq V_{t_1}. \\
&temperature(X, Y_1) : [\{\mathbf{m}\}, V_1, \{V_t\}] \leftarrow \\
&\qquad Y_1 > Y_2 \,\| \\
&\qquad recent\_temperature(X, Y_1) : [\{\mathbf{m}\}, V_1, \{V_t\}] \,\& \\
&\qquad temperature(X, Y_2) : [\{5\}, V_2, \{V_{t_2}\}]. \\
&temperature(X, Y) : [\{\mathbf{m}\}, V_1 \sqcup V_2, \{V_t\}] \leftarrow \\
&\qquad V_t = \underline{\max}(V_{t_1}, V_{t_2}) \,\| \\
&\qquad temperature(X, Y) : [\{3\}, V_1, \{V_{t_1}\}] \,\& \\
&\qquad temperature(X, Y) : [\{4\}, V_2, \{V_{t_2}\}]. \\
&temperature(a, Y) : [\{\mathbf{m}\}, V, \{V_t\}] \leftarrow \\
&\qquad temperature(a, Y) : [\{5\}, V, \{V_{t_1}\}]. \\
&temperature(X, Y) : [\{\mathbf{m}\}, f(V), \{V_t\}] \leftarrow
\end{aligned}
$$

$$temperature(X, Y) : [\{5\}, V, \{V_t\}] \,\&\, X \neq a.$$

$temperature(a, 45) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(b, 60) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(c, 30) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(d, 70) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(e, 43) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(f, 55) : [\{3\}, 1, \{0\}] \leftarrow.$
$temperature(a, 45) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(b, 65) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(c, 40) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(d, 65) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(e, 45) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(f, 55) : [\{4\}, 1, \{5\}] \leftarrow.$
$temperature(a, 45) : [\{5\}, 1, \{6\}] \leftarrow.$
$temperature(b, 75) : [\{5\}, 1, \{6\}] \leftarrow.$
$temperature(c, 45) : [\{5\}, 1, \{6\}] \leftarrow.$
$temperature(d, 65) : [\{5\}, 1, \{6\}] \leftarrow.$
$temperature(e, 50) : [\{5\}, 1, \{6\}] \leftarrow.$
$temperature(f, 58) : [\{5\}, 1, \{6\}] \leftarrow.$