

A Formal Model for Situated Semantic Alignment

Manuel Atencia Marco Schorlemmer
IIIA, Artificial Intelligence Research Institute
CSIC, Spanish National Research Council
Bellaterra (Barcelona), Catalonia, Spain
{manu, marco}@iiia.csic.es

ABSTRACT

Ontology matching is currently a key technology to achieve the semantic alignment of ontological entities used by knowledge-based applications, and therefore to enable their interoperability in distributed environments such as multi-agent systems. Most ontology matching mechanisms, however, assume matching prior integration and rely on semantics that has been coded a priori in concept hierarchies or external sources. In this paper, we present a formal model for a semantic alignment procedure that incrementally aligns differing conceptualisations of two or more agents relative to their respective perception of the environment or domain they are acting in. It hence makes the situation in which the alignment occurs explicit in the model. We resort to Channel Theory to carry out the formalisation.

Categories and Subject Descriptors

I.2.11 [Artificial Intelligence]: Distributed Artificial Intelligence—*coherence and coordination, multiagent systems*;
D.2.12 [Software Engineering]: Interoperability—*data mapping*; I.2.4 [Artificial Intelligence]: Knowledge Representation Formalisms and Methods—*semantic networks, relation systems*.

General Terms

Theory

Keywords

Semantic alignment, distributed logic, channel refinement

1. INTRODUCTION

An ontology is commonly defined as a specification of the conceptualisation of a particular domain. It fixes the vocabulary used by knowledge engineers to denote concepts and their relations, and it constrains the interpretation of this vocabulary to the meaning originally intended by knowledge

engineers. As such, ontologies have been widely adopted as a key technology that may favour knowledge sharing in distributed environments, such as multi-agent systems, federated databases, or the Semantic Web. But the proliferation of many diverse ontologies caused by different conceptualisations of even the same domain—and their subsequent specification using varying terminology—has highlighted the need of ontology matching techniques that are capable of computing semantic relationships between entities of separately engineered ontologies. [5, 11]

Until recently, most ontology matching mechanisms developed so far have taken a classical functional approach to the semantic heterogeneity problem, in which ontology matching is seen as a process taking two or more ontologies as input and producing a semantic alignment of ontological entities as output [3]. Furthermore, matching often has been carried out at design-time, before integrating knowledge-based systems or making them interoperate. This might have been successful for clearly delimited and stable domains and for closed distributed systems, but it is untenable and even undesirable for the kind of applications that are currently deployed in open systems. Multi-agent communication, peer-to-peer information sharing, and web-service composition are all of a decentralised, dynamic, and open-ended nature, and they require ontology matching to be locally performed during run-time. In addition, in many situations peer ontologies are not even open for inspection (e.g., when they are based on commercially confidential information).

Certainly, there exist efforts to efficiently match ontological entities at run-time, taking only those ontology fragments that are necessary for the task at hand [10, 13, 9, 8]. Nevertheless, the techniques used by these systems to establish the semantic relationships between ontological entities—even though applied at run-time—still exploit *a priori* defined concept taxonomies as they are represented in the graph-based structures of the ontologies to be matched, use previously existing external sources such as thesauri (e.g., WordNet) and upper-level ontologies (e.g., CyC or SUMO), or resort to additional background knowledge repositories or shared instances.

We claim that semantic alignment of ontological terminology is ultimately relative to the particular situation in which the alignment is carried out, and that this situation should be made explicit and brought into the alignment mechanism. Even two agents with identical conceptualisation capabilities, and using exactly the same vocabulary to specify their respective conceptualisations may fail to interoperate

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAMAS'07 May 14–18 2007, Honolulu, Hawaii, USA.
Copyright 2007 IFAAMAS.

in a concrete situation because of their differing perception of the domain. Imagine a situation in which two agents are facing each other in front of a checker board. Agent A_1 may conceptualise a figure on the board as situated on the left margin of the board, while agent A_2 may conceptualise the same figure as situated on the right. Although the conceptualisation of ‘left’ and ‘right’ is done in exactly the same manner by both agents, and even if both use the terms *left* and *right* in their communication, they still will need to align their respective vocabularies if they want to successfully communicate to each other actions that change the position of figures on the checker board. Their semantic alignment, however, will only be valid in the scope of their interaction within this particular situation or environment. The same agents situated differently may produce a different alignment.

This scenario is reminiscent to those in which a group of distributed agents adapt to form an ontology and a shared lexicon in an emergent, bottom-up manner, with only local interactions and no central control authority [12]. This sort of self-organised emergence of shared meaning is namely ultimately grounded on the physical interaction of agents with the environment. In this paper, however, we address the case in which agents are already endowed with a top-down engineered ontology (it can even be the same one), which they do not adapt or refine, but for which they want to find the semantic relationships with separate ontologies of other agents on the grounds of their communication within a specific situation. In particular, we provide a formal model that formalises situated semantic alignment as a sequence of information-channel refinements in the sense of Barwise and Seligman’s theory of information flow [1]. This theory is particularly useful for our endeavour because it models the flow of information occurring in distributed systems due to the particular situations—or tokens—that carry information. Analogously, the semantic alignment that will allow information to flow ultimately will be carried by the particular situation agents are acting in.

We shall therefore consider a scenario with two or more agents situated in an environment. Each agent will have its own viewpoint of the environment so that, if the environment is in a concrete state, both agents may have different perceptions of this state. Because of these differences there may be a mismatch in the meaning of the syntactic entities by which agents describe their perceptions (and which constitute the agents’ respective ontologies). We state that these syntactic entities can be related according to the intrinsic semantics provided by the existing relationship between the agents’ viewpoint of the environment. The existence of this relationship is precisely justified by the fact that the agents are situated and observe the same environment.

In Section 2 we describe our formal model for Situated Semantic Alignment (SSA). First, in Section 2.1 we associate a channel to the scenario under consideration and show how the distributed logic generated by this channel provides the logical relationships between the agents’ viewpoints of the environment. Second, in Section 2.2 we present a method by which agents obtain approximations of this distributed logic. These approximations gradually become more reliable as the method is applied. In Section 3 we report on an application of our method. Conclusions and further work are analyzed in Section 4. Finally, an appendix summarizes the terms and theorems of Channel theory used along the paper. We do not

assume any knowledge of Channel Theory; we restate basic definitions and theorems in the appendix, but any detailed exposition of the theory is outside the scope of this paper.

2. A FORMAL MODEL FOR SSA

2.1 The Logic of SSA

Consider a scenario with two agents A_1 and A_2 situated in an environment E (the generalization to any numerable set of agents is straightforward). We associate a numerable set S of states to E and, at any given instant, we suppose E to be in one of these states. We further assume that each agent is able to observe the environment and has its own perception of it. This ability is faithfully captured by a surjective function $see_i : S \rightarrow P_i$, where $i \in \{1, 2\}$, and typically see_1 and see_2 are different.

According to Channel Theory, information is only viable where there is a systematic way of classifying some range of things as being this way or that, in other words, where there is a classification (see appendix A). So in order to be within the framework of Channel Theory, we must associate classifications to the components of our system.

For each $i \in \{1, 2\}$, we consider a classification \mathbf{A}_i that models A_i ’s viewpoint of E . First, $tok(\mathbf{A}_i)$ is composed of A_i ’s perceptions of E states, that is, $tok(\mathbf{A}_i) = P_i$. Second, $typ(\mathbf{A}_i)$ contains the syntactic entities by which A_i describes its perceptions, the ones constituting the ontology of A_i . Finally, $\models_{\mathbf{A}_i}$ synthesizes how A_i relates its perceptions with these syntactic entities.

Now, with the aim of associating environment E with a classification \mathbf{E} we choose the *power classification* of S as \mathbf{E} , which is the classification whose set of types is equal to 2^S , whose tokens are the elements of S , and for which a token e is of type ε if $e \in \varepsilon$. The reason for taking the power classification is because there are no syntactic entities that may play the role of types for \mathbf{E} since, in general, there is no global conceptualisation of the environment. However, the set of types of the power classification includes all possible token configurations potentially described by types. Thus $tok(\mathbf{E}) = S$, $typ(\mathbf{E}) = 2^S$ and $e \models_{\mathbf{E}} \varepsilon$ if and only if $e \in \varepsilon$.

The notion of channel (see appendix A) is fundamental in Barwise and Seligman’s theory. The information flow among the components of a distributed system is modelled in terms of a channel and the relationships among these components are expressed via infomorphisms (see appendix A) which provide a way of moving information between them.

The information flow of the scenario under consideration is accurately described by channel $\mathcal{E} = \{f_i : \mathbf{A}_i \rightarrow \mathbf{E}\}_{i \in \{1, 2\}}$ defined as follows:

- $\hat{f}_i(\alpha) = \{e \in tok(\mathbf{E}) \mid see_i(e) \models_{\mathbf{A}_i} \alpha\}$ for each $\alpha \in typ(\mathbf{A}_i)$
- $\check{f}_i(e) = see_i(e)$ for each $e \in tok(\mathbf{E})$

where $i \in \{1, 2\}$. Definition of \check{f}_i seems natural while \hat{f}_i is defined in such a way that the fundamental property of the infomorphisms is fulfilled:

$$\begin{aligned} \check{f}_i(e) \models_{\mathbf{A}_i} \alpha & \text{ iff } see_i(e) \models_{\mathbf{A}_i} \alpha & (\text{by definition of } \check{f}_i) \\ & \text{ iff } e \in \hat{f}_i(\alpha) & (\text{by definition of } \hat{f}_i) \\ & \text{ iff } e \models_{\mathbf{E}} \hat{f}_i(\alpha) & (\text{by definition of } \models_{\mathbf{E}}) \end{aligned}$$

Consequently, \mathbf{E} is the core of channel \mathcal{E} and a state $e \in tok(\mathbf{E})$ connects agents' perceptions $f_1(e)$ and $f_2(e)$ (see Figure 1).

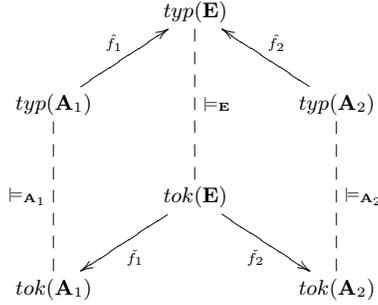


Figure 1: Channel \mathcal{E}

\mathcal{E} explains the information flow of our scenario by virtue of agents A_1 and A_2 being situated and perceiving the same environment E . We want to obtain meaningful relations among agents' syntactic entities, that is, agents' types. We state that meaningfulness must be in accord with \mathcal{E} .

The sum operation (see appendix A) gives us a way of putting the two agents' classifications of channel \mathcal{E} together into a single classification, namely $\mathbf{A}_1 + \mathbf{A}_2$, and also the two infomorphisms together into a single infomorphism, $f_1 + f_2 : \mathbf{A}_1 + \mathbf{A}_2 \rightarrow \mathbf{E}$.

$\mathbf{A}_1 + \mathbf{A}_2$ assembles agents' classifications in a very coarse way. $tok(\mathbf{A}_1 + \mathbf{A}_2)$ is the cartesian product of $tok(\mathbf{A}_1)$ and $tok(\mathbf{A}_2)$, that is, $tok(\mathbf{A}_1 + \mathbf{A}_2) = \{ \langle p_1, p_2 \rangle \mid p_i \in P_i \}$, so a token of $\mathbf{A}_1 + \mathbf{A}_2$ is a pair of agents' perceptions with no restrictions. $typ(\mathbf{A}_1 + \mathbf{A}_2)$ is the disjoint union of $typ(\mathbf{A}_1)$ and $typ(\mathbf{A}_2)$, and $\langle p_1, p_2 \rangle$ is of type $\langle i, \alpha \rangle$ if p_i is of type α . We attach importance to take the disjoint union because A_1 and A_2 could use identical types with the purpose of describing their respective perceptions of E .

Classification $\mathbf{A}_1 + \mathbf{A}_2$ seems to be the natural place in which to search for relations among agents' types. Now, Channel Theory provides a way to make all these relations explicit in a logical fashion by means of theories and local logics (see appendix A). The theory generated by the sum classification, $Th(\mathbf{A}_1 + \mathbf{A}_2)$, and hence its logic generated, $Log(\mathbf{A}_1 + \mathbf{A}_2)$, involve all those constraints among agents' types valid according to $\mathbf{A}_1 + \mathbf{A}_2$. Notice however that these constraints are obvious. As we stated above, meaningfulness must be in accord with channel \mathcal{E} .

Classifications $\mathbf{A}_1 + \mathbf{A}_2$ and \mathbf{E} are connected via the sum infomorphism, $f = f_1 + f_2$, where:

- $\hat{f}(\langle i, \alpha \rangle) = \hat{f}_i(\alpha) = \{ e \in tok(\mathbf{E}) \mid see_i(e) \models_{\mathbf{A}_i} \alpha \}$ for each $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$
- $\check{f}(e) = \langle \check{f}_1(e), \check{f}_2(e) \rangle = \langle see_1(e), see_2(e) \rangle$ for each $e \in tok(\mathbf{E})$

Meaningful constraints among agents' types are in accord with channel \mathcal{E} because they are computed making use of f as we expound below.

As important as the notion of channel is the concept of distributed logic (see appendix A). Given a channel \mathcal{C} and a logic \mathcal{L} on its core, $DLog_{\mathcal{C}}(\mathcal{L})$ represents the reasoning

about relations among the components of \mathcal{C} justified by \mathcal{L} . If $\mathcal{L} = Log(\mathcal{C})$, the distributed logic, we denoted by $Log(\mathcal{C})$, captures in a logical fashion the information flow inherent in the channel.

In our case, $Log(\mathcal{E})$ explains the relationship between the agents' viewpoints of the environment in a logical fashion. On the one hand, constraints of $Th(Log(\mathcal{E}))$ are defined by:

$$\Gamma \vdash_{Log(\mathcal{E})} \Delta \quad \text{if} \quad \hat{f}[\Gamma] \vdash_{Log(\mathbf{E})} \hat{f}[\Delta] \quad (1)$$

where $\Gamma, \Delta \subseteq typ(\mathbf{A}_1 + \mathbf{A}_2)$. On the other hand, the set of normal tokens, $N_{Log(\mathcal{E})}$, is equal to the range of function \check{f} :

$$\begin{aligned} N_{Log(\mathcal{E})} &= \check{f}[tok(\mathbf{E})] \\ &= \{ \langle see_1(e), see_2(e) \rangle \mid e \in tok(\mathbf{E}) \} \end{aligned}$$

Therefore, a normal token is a pair of agents' perceptions that are restricted by coming from the same environment state (unlike $\mathbf{A}_1 + \mathbf{A}_2$ tokens).

All constraints of $Th(Log(\mathcal{E}))$ are satisfied by all normal tokens (because of being a logic). In this particular case, this condition is also sufficient (the proof is straightforward); as alternative to (1) we have:

$$\begin{aligned} \Gamma \vdash_{Log(\mathcal{E})} \Delta \quad \text{iff} \quad & \text{for all } e \in tok(\mathbf{E}), \\ & \text{if } (\forall (i, \gamma) \in \Gamma) [see_i(e) \models_{\mathbf{A}_i} \gamma] \\ & \text{then } (\exists (j, \delta) \in \Delta) [see_j(e) \models_{\mathbf{A}_j} \delta] \end{aligned} \quad (2)$$

where $\Gamma, \Delta \subseteq typ(\mathbf{A}_1 + \mathbf{A}_2)$.

$Log(\mathcal{E})$ is the logic of SSA. $Th(Log(\mathcal{E}))$ comprises the most meaningful constraints among agents' types in accord with channel \mathcal{E} . In other words, the logic of SSA contains and also justifies the most meaningful relations among those syntactic entities that agents use in order to describe their own environment perceptions.

$Log(\mathcal{E})$ is complete since $Log(\mathbf{E})$ is complete but it is not necessarily sound because although $Log(\mathbf{E})$ is sound, \check{f} is not surjective in general (see appendix B). If $Log(\mathcal{E})$ is also sound then $Log(\mathcal{E}) = Log(\mathbf{A}_1 + \mathbf{A}_2)$ (see appendix B). That means there is no significant relation between agents' points of view of the environment according to \mathcal{E} . It is just the fact that $Log(\mathcal{E})$ is unsound what allows a significant relation between the agents' viewpoints. This relation is expressed at the type level in terms of constraints by $Th(Log(\mathcal{E}))$ and at the token level by $N_{Log(\mathcal{E})}$.

2.2 Approaching the logic of SSA through communication

We have dubbed $Log(\mathcal{E})$ the logic of SSA. $Th(Log(\mathcal{E}))$ comprehends the most meaningful constraints among agents' types according to \mathcal{E} . The problem is that neither agent can make use of this theory because they do not know \mathcal{E} completely. In this section, we present a method by which agents obtain approximations to $Th(Log(\mathcal{E}))$. We also prove these approximations gradually become more reliable as the method is applied.

Agents can obtain approximations to $Th(Log(\mathcal{E}))$ through communication. A_1 and A_2 communicate by exchanging information about their perceptions of environment states. This information is expressed in terms of their own classification relations. Specifically, if E is in a concrete state e , we assume that agents can convey to each other which types are satisfied by their respective perceptions of e and which are not. This exchange generates a channel $\mathcal{C} = \{ f_i : \mathbf{A}_i \rightarrow$

$\mathbf{C}\}_{i \in \{1,2\}}$ and $Th(Log(\mathbf{C}))$ contains the constraints among agents' types justified by the fact that agents have observed e . Now, if E turns to another state e' and agents proceed as before, another channel $\mathbf{C}' = \{f'_i : \mathbf{A}_i \rightarrow \mathbf{C}'\}_{i \in \{1,2\}}$ gives account of the new situation considering also the previous information. $Th(Log(\mathbf{C}'))$ comprises the constraints among agents' types justified by the fact that agents have observed e and e' . The significant point is that \mathbf{C}' is a refinement of \mathbf{C} (see appendix A). Theorem 2.1 below ensures that the refined channel involves more reliable information.

The communication supposedly ends when agents have observed all the environment states. Again this situation can be modeled by a channel, call it $\mathbf{C}^* = \{f_i^* : \mathbf{A}_i \rightarrow \mathbf{C}^*\}_{i \in \{1,2\}}$. Theorem 2.2 states that $Th(Log(\mathbf{C}^*)) = Th(Log(\mathcal{E}))$.

Theorem 2.1 and Theorem 2.2 assure that applying the method agents can obtain approximations to $Th(Log(\mathcal{E}))$ gradually more reliable.

THEOREM 2.1. *Let $\mathbf{C} = \{f_i : \mathbf{A}_i \rightarrow \mathbf{C}\}_{i \in \{1,2\}}$ and $\mathbf{C}' = \{f'_i : \mathbf{A}_i \rightarrow \mathbf{C}'\}_{i \in \{1,2\}}$ be two channels. If \mathbf{C}' is a refinement of \mathbf{C} then:*

1. $Th(Log(\mathbf{C}')) \subseteq Th(Log(\mathbf{C}))$
2. $N_{Log(\mathbf{C}')} \supseteq N_{Log(\mathbf{C})}$

PROOF. Since \mathbf{C}' is a refinement of \mathbf{C} then there exists a refinement infomorphism r from \mathbf{C}' to \mathbf{C} ; so $f_i = r \circ f'_i$. Let $\mathbf{A} =_{def} \mathbf{A}_1 + \mathbf{A}_2$, $f =_{def} f_1 + f_2$ and $f' =_{def} f'_1 + f'_2$.

1. Let Γ and Δ be subsets of $typ(\mathbf{A})$ and assume that $\Gamma \vdash_{Log(\mathbf{C}')} \Delta$, which means $\hat{f}'[\Gamma] \vdash_{\mathbf{C}'} \hat{f}'[\Delta]$. We have to prove $\Gamma \vdash_{Log(\mathbf{C})} \Delta$, or equivalently, $\hat{f}[\Gamma] \vdash_{\mathbf{C}} \hat{f}[\Delta]$. We proceed by reductio ad absurdum. Suppose $c \in tok(\mathbf{C})$ does not satisfy the sequent $\langle \hat{f}[\Gamma], \hat{f}[\Delta] \rangle$. Then $c \models_{\mathbf{C}} \hat{f}(\gamma)$ for all $\gamma \in \Gamma$ and $c \not\models_{\mathbf{C}} \hat{f}(\delta)$ for all $\delta \in \Delta$. Let us choose an arbitrary $\gamma \in \Gamma$. We have that $\gamma = \langle i, \alpha \rangle$ for some $\alpha \in typ(\mathbf{A}_i)$ and $i \in \{1,2\}$. Thus $\hat{f}(\gamma) = \hat{f}(\langle i, \alpha \rangle) = \hat{f}_i(\alpha) = \hat{r} \circ \hat{f}'_i(\alpha) = \hat{r}(\hat{f}'_i(\alpha))$. Therefore:

$$\begin{aligned} c \models_{\mathbf{C}} \hat{f}(\gamma) & \text{ iff } c \models_{\mathbf{C}} \hat{r}(\hat{f}'_i(\alpha)) \\ & \text{ iff } \check{r}(c) \models_{\mathbf{C}'} \hat{f}'_i(\alpha) \\ & \text{ iff } \check{r}(c) \models_{\mathbf{C}'} \hat{f}'(\langle i, \alpha \rangle) \\ & \text{ iff } \check{r}(c) \models_{\mathbf{C}'} \hat{f}'(\gamma) \end{aligned}$$

Consequently, $\check{r}(c) \models_{\mathbf{C}'} \hat{f}'(\gamma)$ for all $\gamma \in \Gamma$. Since $\hat{f}'[\Gamma] \vdash_{\mathbf{C}'} \hat{f}'[\Delta]$ then there exists $\delta^* \in \Delta$ such that $\check{r}(c) \models_{\mathbf{C}'} \hat{f}'(\delta^*)$. A sequence of equivalences similar to the above one justifies $c \models_{\mathbf{C}} \hat{f}(\delta^*)$, contradicting that c is a counterexample to $\langle \hat{f}[\Gamma], \hat{f}[\Delta] \rangle$. Hence $\Gamma \vdash_{Log(\mathbf{C})} \Delta$ as we wanted to prove.

2. Let $\langle a_1, a_2 \rangle \in tok(\mathbf{A})$ and assume $\langle a_1, a_2 \rangle \in N_{Log(\mathbf{C})}$. Therefore, there exists c token in \mathbf{C} such that $\langle a_1, a_2 \rangle = \hat{f}(c)$. Then we have $a_i = \hat{f}_i(c) = \hat{f}'_i \circ \check{r}(c) = \hat{f}'_i(\check{r}(c))$, for $i \in \{1,2\}$. Hence $\langle a_1, a_2 \rangle = \hat{f}'(\check{r}(c))$ and $\langle a_1, a_2 \rangle \in N_{Log(\mathbf{C}')}.$ Consequently, $N_{Log(\mathbf{C}')} \supseteq N_{Log(\mathbf{C})}$ which concludes the proof.

□

REMARK 2.1. *Theorem 2.1 asserts that the more refined channel gives more reliable information. Even though its theory has less constraints, it has more normal tokens to which they apply.*

In the remainder of the section, we explicitly describe the process of communication and we conclude with the proof of Theorem 2.2.

Let us assume that $typ(\mathbf{A}_i)$ is finite for $i \in \{1,2\}$ and S is infinite numerable, though the finite case can be treated in a similar form. We also choose an infinite numerable set of symbols $\{c^n \mid n \in \mathbb{N}\}^1$.

We omit infomorphisms superscripts when no confusion arises. Types are usually denoted by greek letters and tokens by latin letters so if f is an infomorphism, $f(\alpha) \equiv \hat{f}(\alpha)$ and $f(a) \equiv \hat{f}(a)$.

Agents communication starts from the observation of E . Let us suppose that E is in state $e^1 \in S = tok(\mathbf{E})$. A_1 's perception of e^1 is $f_1(e^1)$ and A_2 's perception of e^1 is $f_2(e^1)$. We take for granted that A_1 can communicate A_2 those types that are and are not satisfied by $f_1(e^1)$ according to its classification \mathbf{A}_1 . So can A_2 do. Since both $typ(\mathbf{A}_1)$ and $typ(\mathbf{A}_2)$ are finite, this process eventually finishes. After this communication a channel $\mathbf{C}^1 = \{f_i^1 : \mathbf{A}_i \rightarrow \mathbf{C}^1\}_{i=1,2}$ arises (see Figure 2).

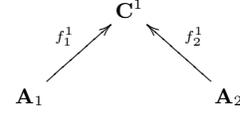


Figure 2: The first communication stage

On the one hand, \mathbf{C}^1 is defined by:

- $tok(\mathbf{C}^1) = \{c^1\}$
- $typ(\mathbf{C}^1) = typ(\mathbf{A}_1 + \mathbf{A}_2)$
- $c^1 \models_{\mathbf{C}^1} \langle i, \alpha \rangle$ if $f_i(e^1) \models_{\mathbf{A}_i} \alpha$
(for every $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$)

On the other hand, f_i^1 , with $i \in \{1,2\}$, is defined by:

- $f_i^1(\alpha) = \langle i, \alpha \rangle$
(for every $\alpha \in typ(\mathbf{A}_i)$)
- $f_i^1(c^1) = f_i(e^1)$

$Log(\mathbf{C}^1)$ represents the reasoning about the first stage of communication. It is easy to prove that $Th(Log(\mathbf{C}^1)) = Th(\mathbf{C}^1)$. The significant point is that both agents know \mathbf{C}^1 as the result of the communication. Hence they can compute separately theory $Th(\mathbf{C}^1) = \langle typ(\mathbf{C}^1), \vdash_{\mathbf{C}^1} \rangle$ which contains the constraints among agents' types justified by the fact that agents have observed e^1 .

Now, let us assume that E turns to a new state e^2 . Agents can proceed as before, exchanging this time information about their perceptions of e^2 . Another channel $\mathbf{C}^2 = \{f_i^2 : \mathbf{A}_i \rightarrow \mathbf{C}^2\}_{i \in \{1,2\}}$ comes up. We define \mathbf{C}^2 so as to take also into account the information provided by the previous stage of communication.

On the one hand, \mathbf{C}^2 is defined by:

- $tok(\mathbf{C}^2) = \{c^1, c^2\}$

¹We write these symbols with superindices because we limit the use of subindices for what concerns to agents. Note this set is chosen with the same cardinality of S .

- $typ(\mathbf{C}^2) = typ(\mathbf{A}_1 + \mathbf{A}_2)$
- $c^k \models_{\mathbf{C}^2} \langle i, \alpha \rangle$ if $f_i(e^k) \models_{\mathbf{A}_i} \alpha$
(for every $k \in \{1, 2\}$ and $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$)

On the other hand, f_i^2 , with $i \in \{1, 2\}$, is defined by:

- $f_i^2(\alpha) = \langle i, \alpha \rangle$
(for every $\alpha \in typ(\mathbf{A}_i)$)
- $f_i^2(c^k) = f_i(e^k)$
(for every $k \in \{1, 2\}$)

$Log(\mathbf{C}^2)$ represents the reasoning about the former and the later communication stages. $Th(Log(\mathbf{C}^2))$ is equal to $Th(\mathbf{C}^2) = \langle typ(\mathbf{C}^2), \vdash_{\mathbf{C}^2} \rangle$, then it contains the constraints among agents' types justified by the fact that agents have observed e^1 and e^2 . A_1 and A_2 knows \mathbf{C}^2 so they can use these constraints. The key point is that channel \mathbf{C}^2 is a refinement of \mathbf{C}^1 . It is easy to check that f^1 defined as the identity function on types and the inclusion function on tokens is a refinement infomorphism (see at the bottom of Figure 3). By Theorem 2.1, \mathbf{C}^2 constraints are more reliable than \mathbf{C}^1 constraints.

In the general situation, once the states e^1, e^2, \dots, e^{n-1} ($n \geq 2$) have been observed and a new state e^n appears, channel $\mathbf{C}^n = \{f_i^n : \mathbf{A}_i \rightarrow \mathbf{C}^n\}_{i \in \{1, 2\}}$ informs about agents communication up to that moment. \mathbf{C}^n definition is similar to the previous ones and analogous remarks can be made (see at the top of Figure 3). Theory $Th(Log(\mathbf{C}^n)) = Th(\mathbf{C}^n) = \langle typ(\mathbf{C}^n), \vdash_{\mathbf{C}^n} \rangle$ contains the constraints among agents' types justified by the fact that agents have observed e^1, e^2, \dots, e^n .

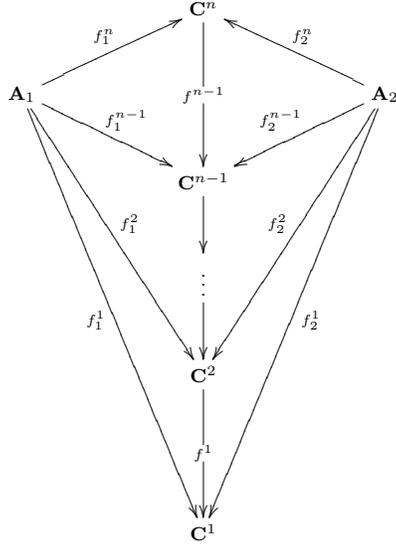


Figure 3: Agents communication

Remember we have assumed that S is infinite numerable. It is therefore unpractical to let communication finish when all environment states have been observed by A_1 and A_2 . At that point, the family of channels $\{\mathbf{C}^n\}_{n \in \mathbb{N}}$ would inform of all the communication stages. It is therefore up to the

agents to decide when to stop communicating should a good enough approximation have been reached for the purposes of their respective tasks. But the study of possible termination criteria is outside the scope of this paper and left for future work. From a theoretical point of view, however, we can consider the channel $\mathbf{C}^* = \{f_i^* : \mathbf{A}_i \rightarrow \mathbf{C}^*\}_{i \in \{1, 2\}}$ which informs of the end of the communication after observing all environment states.

On the one hand, \mathbf{C}^* is defined by:

- $tok(\mathbf{C}^*) = \{c^n \mid n \in \mathbb{N}\}$
- $typ(\mathbf{C}^*) = typ(\mathbf{A}_1 + \mathbf{A}_2)$
- $c^n \models_{\mathbf{C}^*} \langle i, \alpha \rangle$ if $f_i(e^n) \models_{\mathbf{A}_i} \alpha$
(for $n \in \mathbb{N}$ and $\langle i, \alpha \rangle \in typ(\mathbf{A}_1 + \mathbf{A}_2)$)

On the other hand, f_i^* , with $i \in \{1, 2\}$, is defined by:

- $f_i^*(\alpha) = \langle i, \alpha \rangle$
(for $\alpha \in typ(\mathbf{A}_i)$)
- $f_i^*(c^n) = f_i(e^n)$
(for $n \in \mathbb{N}$)

Theorem below constitutes the cornerstone of the model exposed in this paper. It ensures, together with Theorem 2.1, that at each communication stage agents obtain a theory that approximates more closely to the theory generated by the logic of SSA.

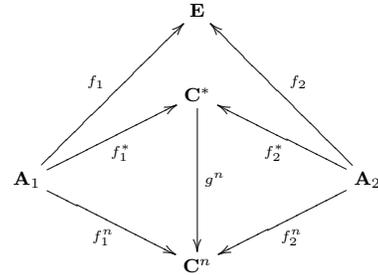
THEOREM 2.2. *The following statements hold:*

1. For all $n \in \mathbb{N}$, \mathbf{C}^* is a refinement of \mathbf{C}^n .
2. $Th(Log(\mathcal{E})) = Th(\mathbf{C}^*) = Th(Log(\mathbf{C}^*))$.

PROOF.

1. It is easy to prove that for each $n \in \mathbb{N}$, g^n defined as the identity function on types and the inclusion function on tokens is a refinement infomorphism from \mathbf{C}^* to \mathbf{C}^n .
2. The second equality is straightforward; the first one follows directly from:

$$\begin{aligned}
 c^n \models_{\mathbf{C}^*} \langle i, \alpha \rangle &\text{ iff } \hat{f}_i(e^n) \models_{\mathbf{A}_i} \alpha \\
 &\text{ (by definition of } \models_{\mathbf{C}^*} \text{)} \\
 &\text{ iff } e^n \models_{\mathbf{E}} \hat{f}_i(\alpha) \\
 &\text{ (because } \hat{f}_i \text{ is infomorphism)} \\
 &\text{ iff } e^n \models_{\mathbf{E}} \hat{f}(\langle i, \alpha \rangle) \\
 &\text{ (by definition of } \hat{f} \text{)}
 \end{aligned}$$



□

3. AN EXAMPLE

In the previous section we have described in great detail our formal model for SSA. However, we have not tackled the practical aspect of the model yet. In this section, we give a brushstroke of the pragmatic view of our approach. We study a very simple example and explain how agents can use those approximations of the logic of SSA they can obtain through communication.

Let us reflect on a system consisting of robots located in a two-dimensional grid looking for packages with the aim of moving them to a certain destination (Figure 4). Robots can carry only one package at a time and they can not move through a package.

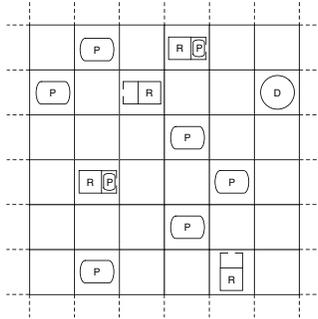


Figure 4: The scenario

Robots have a partial view of the domain and there exist two kinds of robots according to the visual field they have. Some robots are capable of observing the eight adjoining squares but others just observe the three squares they have in front (see Figure 5). We call them URDL (shortened form of Up-Right-Down-Left) and LCR (abbreviation for Left-Center-Right) robots respectively.

Describing the environment states as well as the robots' perception functions is rather tedious and even unnecessary. We assume the reader has all those descriptions in mind.

All robots in the system must be able to solve package distribution problems cooperatively by communicating their intentions to each other. In order to communicate, agents send messages using some ontology. In our scenario, there coexist two ontologies, the UDRL and LCR ontologies. Both of them are very simple and are just confined to describe what robots observe.

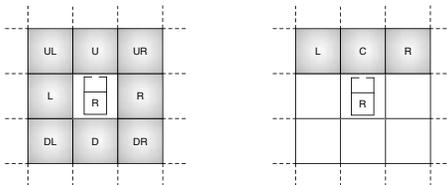


Figure 5: Robots field of vision

When a robot carrying a package finds another package obstructing its way, it can either go around it or, if there is

another robot in its visual field, ask it for assistance. Let us suppose two URDL robots are in a situation like the one depicted in Figure 6. Robot1 (the one carrying a package) decides to ask Robot2 for assistance and sends a request. This request is written below as a KQML message and it should be interpreted intuitively as: *Robot2, pick up the package located in my "Up" square, knowing that you are located in my "Up-Right" square.*

```
(request
:sender      Robot1
:receiver    Robot2
:language    Packages distribution-language
:ontology    URDL-ontology
:content     (PICK UP U(Package) BECAUSE UR(Robot2))
```

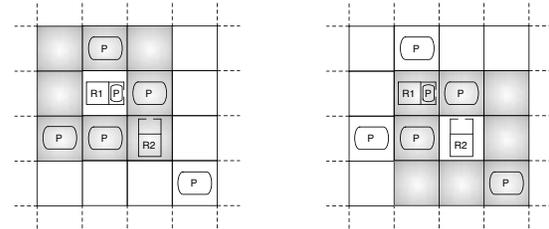


Figure 6: Robot assistance

Robot2 understands the content of the request and it can use a rule represented by the following constraint:

$$\langle 1, UR(Robot2) \rangle, \langle 2, UL(Robot1) \rangle, \langle 1, U(Package) \rangle \\ \vdash \langle 2, U(Package) \rangle$$

The above constraint should be interpreted intuitively as: *if Robot2 is situated in Robot1's "Up-Right" square, Robot1 is situated in Robot2's "Up-Left" square and a package is located in Robot1's "Up" square, then a package is located in Robot2's "Up" square.*

Now, problems arise when a LCR robot and a URDL robot try to interoperate. See Figure 7. Robot1 sends a request of the form:

```
(request
:sender      Robot1
:receiver    Robot2
:language    Packages distribution-language
:ontology    LCR-ontology
:content     (PICK UP R(Robot2) BECAUSE C(Package))
```

Robot2 does not understand the content of the request but they decide to begin a process of alignment -corresponding with a channel C^1 . Once finished, Robot2 searches in $Th(C^1)$ for constraints similar to the expected one, that is, those of the form:

$$\langle 1, R(Robot2) \rangle, \langle 2, UL(Robot1) \rangle, \langle 1, C(Package) \rangle \\ \vdash_{C^1} \langle 2, \lambda(Package) \rangle$$

where $\lambda \in \{U, R, D, L, UR, DR, DL, UL\}$. From these, only the following constraints are plausible according to C^1 :

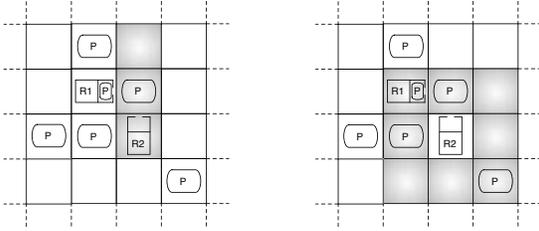


Figure 7: Ontology mismatch

$$\langle 1, R(\text{Robot2}), \langle 2, UL(\text{Robot1}), \langle 1, C(\text{Package}) \rangle \rangle \\ \vdash_{\mathcal{C}^1} \langle 2, U(\text{Package}) \rangle$$

$$\langle 1, R(\text{Robot2}), \langle 2, UL(\text{Robot1}), \langle 1, C(\text{Package}) \rangle \rangle \\ \vdash_{\mathcal{C}^1} \langle 2, L(\text{Package}) \rangle$$

$$\langle 1, R(\text{Robot2}), \langle 2, UL(\text{Robot1}), \langle 1, C(\text{Package}) \rangle \rangle \\ \vdash_{\mathcal{C}^1} \langle 2, DR(\text{Package}) \rangle$$

If subsequently both robots adopting the same roles take part in a situation like the one depicted in Figure 8, a new process of alignment -corresponding with a channel \mathcal{C}^2 - takes place. \mathcal{C}^2 also considers the previous information and hence refines \mathcal{C}^1 . The only constraint from the above ones that remains plausible according to \mathcal{C}^2 is :

$$\langle 1, R(\text{Robot2}), \langle 2, UL(\text{Robot1}), \langle 1, C(\text{Package}) \rangle \rangle \\ \vdash_{\mathcal{C}^2} \langle 2, U(\text{Package}) \rangle$$

Notice that this constraint is an element of the theory of the distributed logic. Agents communicate in order to cooperate successfully and success is guaranteed using constrains of the distributed logic.

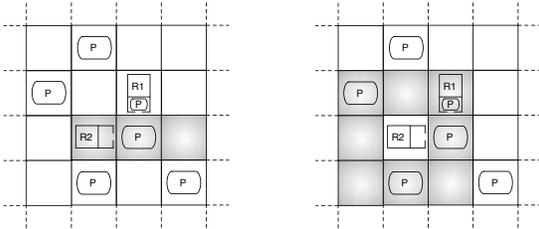


Figure 8: Refinement

4. CONCLUSIONS AND FURTHER WORK

In this paper we have exposed a formal model of semantic alignment as a sequence of information-channel refinements that are relative to the particular states of the environment in which two agents communicate and align their respective conceptualisations of these states. Before us, Kent [6] and Kalfoglou and Schorlemmer [4, 10] have applied Channel Theory to formalise semantic alignment using also Barwise

and Seligman's insight to focus on tokens as the enablers of information flow. Their approach to semantic alignment, however, like most ontology matching mechanisms developed to date (regardless of whether they follow a functional, design-time-based approach, or an interaction-based, run-time-based approach), still defines semantic alignment in terms of *a priori* design decisions such as the concept taxonomy of the ontologies or the external sources brought into the alignment process. Instead the model we have presented in this paper makes explicit the particular states of the environment in which agents are situated and are attempting to gradually align their ontological entities.

In the future, our effort will focus on the practical side of the situated semantic alignment problem. We plan to further refine the model presented here (e.g., to include pragmatic issues such as termination criteria for the alignment process) and to devise concrete ontology negotiation protocols based on this model that agents may be able to enact. The formal model exposed in this paper will constitute a solid base of future practical results.

Acknowledgements

This work is supported under the UPIC project, sponsored by Spain's Ministry of Education and Science under grant number TIN2004-07461-C02-02 and also under the Open-Knowledge Specific Targeted Research Project (STREP), sponsored by the European Commission under contract number FP6-027253. Marco Schorlemmer is supported by a *Ramón y Cajal* Research Fellowship from Spain's Ministry of Education and Science, partially funded by the European Social Fund.

5. REFERENCES

- [1] J. Barwise and J. Seligman. *Information Flow: The Logic of Distributed Systems*. Cambridge University Press, 1997.
- [2] C. Ghidini and F. Giunchiglia. Local models semantics, or contextual reasoning = locality + compatibility. *Artificial Intelligence*, 127(2):221–259, 2001.
- [3] F. Giunchiglia and P. Shvaiko. Semantic matching. *The Knowledge Engineering Review*, 18(3):265–280, 2004.
- [4] Y. Kalfoglou and M. Schorlemmer. IF-Map: An ontology-mapping method based on information-flow theory. In *Journal on Data Semantics I*, LNCS 2800, 2003.
- [5] Y. Kalfoglou and M. Schorlemmer. Ontology mapping: The state of the art. *The Knowledge Engineering Review*, 18(1):1–31, 2003.
- [6] R. E. Kent. Semantic integration in the Information Flow Framework. In *Semantic Interoperability and Integration*, Dagstuhl Seminar Proceedings 04391, 2005.
- [7] D. Lenat. CyC: A large-scale investment in knowledge infrastructure. *Communications of the ACM*, 38(11), 1995.
- [8] V. López, M. Sabou, and E. Motta. PowerMap: Mapping the real Semantic Web on the fly. *Proceedings of the ISWC'06*, 2006.
- [9] F. McNeill. *Dynamic Ontology Refinement*. PhD

thesis, School of Informatics, The University of Edinburgh, 2006.

- [10] M. Schorlemmer and Y. Kalfoglou. Progressive ontology alignment for meaning coordination: An information-theoretic foundation. In *4th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 2005.
- [11] P. Shvaiko and J. Euzenat. A survey of schema-based matching approaches. In *Journal on Data Semantics IV*, LNCS 3730, 2005.
- [12] L. Steels. The Origins of Ontologies and Communication Conventions in Multi-Agent Systems. In *Journal of Autonomous Agents and Multi-Agent Systems*, 1(2), 169-194, 1998.
- [13] J. van Diggelen et al. ANEMONE: An Effective Minimal Ontology Negotiation Environment In *5th Int. Joint Conf. on Autonomous Agents and Multiagent Systems*, 2006

APPENDIX

A. CHANNEL THEORY TERMS

Classification: is a tuple $\mathbf{A} = \langle tok(\mathbf{A}), typ(\mathbf{A}), \models_{\mathbf{A}} \rangle$ where $tok(\mathbf{A})$ is a set of *tokens*, $typ(\mathbf{A})$ is a set of *types* and $\models_{\mathbf{A}}$ is a binary relation between $tok(\mathbf{A})$ and $typ(\mathbf{A})$. If $a \models_{\mathbf{A}} \alpha$ then a is said to be of type α .

Infomorphism: $f : \mathbf{A} \rightarrow \mathbf{B}$ from classifications \mathbf{A} to \mathbf{B} is a contravariant pair of functions $f = \langle \hat{f}, \check{f} \rangle$, where $\hat{f} : typ(\mathbf{A}) \rightarrow typ(\mathbf{B})$ and $\check{f} : tok(\mathbf{B}) \rightarrow tok(\mathbf{A})$, satisfying the following fundamental property:

$$\check{f}(b) \models_{\mathbf{A}} \alpha \text{ iff } b \models_{\mathbf{B}} \hat{f}(\alpha)$$

for each token $b \in tok(\mathbf{B})$ and each type $\alpha \in typ(\mathbf{A})$.

Channel: consists of two infomorphisms $\mathcal{C} = \{f_i : \mathbf{A}_i \rightarrow \mathbf{C}\}_{i \in \{1,2\}}$ with a common codomain \mathbf{C} , called the *core* of \mathcal{C} . \mathbf{C} tokens are called *connections* and a connection c is said to *connect* tokens $\check{f}_1(c)$ and $\check{f}_2(c)$.²

Sum: given classifications \mathbf{A} and \mathbf{B} , the *sum* of \mathbf{A} and \mathbf{B} , denoted by $\mathbf{A} + \mathbf{B}$, is the classification with $tok(\mathbf{A} + \mathbf{B}) = tok(\mathbf{A}) \times tok(\mathbf{B}) = \{ \langle a, b \rangle \mid a \in tok(\mathbf{A}) \text{ and } b \in tok(\mathbf{B}) \}$, $typ(\mathbf{A} + \mathbf{B}) = typ(\mathbf{A}) \uplus typ(\mathbf{B}) = \{ \langle i, \gamma \rangle \mid i = 1 \text{ and } \gamma \in typ(\mathbf{A}) \text{ or } i = 2 \text{ and } \gamma \in typ(\mathbf{B}) \}$ and relation $\models_{\mathbf{A} + \mathbf{B}}$ defined by:

$$\begin{aligned} \langle a, b \rangle \models_{\mathbf{A} + \mathbf{B}} \langle 1, \alpha \rangle & \text{ if } a \models_{\mathbf{A}} \alpha \\ \langle a, b \rangle \models_{\mathbf{A} + \mathbf{B}} \langle 2, \beta \rangle & \text{ if } b \models_{\mathbf{B}} \beta \end{aligned}$$

Given infomorphisms $f : \mathbf{A} \rightarrow \mathbf{C}$ and $g : \mathbf{B} \rightarrow \mathbf{C}$, the sum $f + g : \mathbf{A} + \mathbf{B} \rightarrow \mathbf{C}$ is defined on types by $(f + g)(\langle 1, \alpha \rangle) = \hat{f}(\alpha)$ and $(f + g)(\langle 2, \beta \rangle) = \hat{g}(\beta)$, and on tokens by $(f + g)(c) = \langle \check{f}(c), \check{g}(c) \rangle$.

Theory: given a set Σ , a *sequent* of Σ is a pair $\langle \Gamma, \Delta \rangle$ of subsets of Σ . A binary relation \vdash between subsets of Σ is called a *consequence relation* on Σ . A *theory* is a pair $T = \langle \Sigma, \vdash \rangle$ where \vdash is a consequence relation on Σ . A sequent $\langle \Gamma, \Delta \rangle$ of Σ for which $\Gamma \vdash \Delta$ is called a *constraint* of the theory T . T is *regular* if it satisfies:

1. *Identity:* $\alpha \vdash \alpha$
2. *Weakening:* if $\Gamma \vdash \Delta$, then $\Gamma, \Gamma' \vdash \Delta, \Delta'$

²In fact, this is the definition of a binary channel. A channel can be defined with an arbitrary index set.

3. *Global Cut:* if $\Gamma, \Pi_0 \vdash \Delta, \Pi_1$ for each partition $\langle \Pi_0, \Pi_1 \rangle$ of Π (i.e., $\Pi_0 \cup \Pi_1 = \Pi$ and $\Pi_0 \cap \Pi_1 = \emptyset$), then $\Gamma \vdash \Delta$

for all $\alpha \in \Sigma$ and all $\Gamma, \Gamma', \Delta, \Delta', \Pi \subseteq \Sigma$.³

Theory generated by a classification: let \mathbf{A} be a classification. A token $a \in tok(\mathbf{A})$ *satisfies* a sequent $\langle \Gamma, \Delta \rangle$ of $typ(\mathbf{A})$ provided that if a is of every type in Γ then it is of some type in Δ . The *theory generated* by \mathbf{A} , denoted by $Th(\mathbf{A})$, is the theory $\langle typ(\mathbf{A}), \vdash_{\mathbf{A}} \rangle$ where $\Gamma \vdash_{\mathbf{A}} \Delta$ if every token in \mathbf{A} satisfies $\langle \Gamma, \Delta \rangle$.

Local logic: is a tuple $\mathcal{L} = \langle tok(\mathcal{L}), typ(\mathcal{L}), \models_{\mathcal{L}}, \vdash_{\mathcal{L}}, N_{\mathcal{L}} \rangle$ where:

1. $\langle tok(\mathcal{L}), typ(\mathcal{L}), \models_{\mathcal{L}} \rangle$ is a classification denoted by $Cl_{\mathcal{L}}(\mathcal{L})$,
2. $\langle typ(\mathcal{L}), \vdash_{\mathcal{L}} \rangle$ is a regular theory denoted by $Th(\mathcal{L})$,
3. $N_{\mathcal{L}}$ is a subset of $tok(\mathcal{L})$, called the *normal tokens* of \mathcal{L} , which satisfy all constraints of $Th(\mathcal{L})$.

A local logic \mathcal{L} is *sound* if every token in $Cl_{\mathcal{L}}(\mathcal{L})$ is normal, that is, $N_{\mathcal{L}} = tok(\mathcal{L})$. \mathcal{L} is *complete* if every sequent of $typ(\mathcal{L})$ satisfied by every normal token is a constraint of $Th(\mathcal{L})$.

Local logic generated by a classification: given a classification \mathbf{A} , the *local logic generated* by \mathbf{A} , written $Log(\mathbf{A})$, is the local logic on \mathbf{A} (i.e., $Cl_{Log(\mathbf{A})} = \mathbf{A}$), with $Th(Log(\mathbf{A})) = Th(\mathbf{A})$ and such that all its tokens are normal, i.e., $N_{Log(\mathbf{A})} = tok(\mathbf{A})$.

Inverse image: given an infomorphism $f : \mathbf{A} \rightarrow \mathbf{B}$ and a local logic \mathcal{L} on \mathbf{B} , the *inverse image* of \mathcal{L} under f , denoted $f^{-1}[\mathcal{L}]$, is the local logic on \mathbf{A} such that $\Gamma \vdash_{f^{-1}[\mathcal{L}]} \Delta$ if $\hat{f}[\Gamma] \vdash_{\mathcal{L}} \hat{f}[\Delta]$ and $N_{f^{-1}[\mathcal{L}]} = \hat{f}[N_{\mathcal{L}}] = \{ a \in tok(\mathbf{A}) \mid a = \check{f}(b) \text{ for some } b \in N_{\mathcal{L}} \}$.

Distributed logic: let $\mathcal{C} = \{f_i : \mathbf{A}_i \rightarrow \mathbf{C}\}_{i \in \{1,2\}}$ be a channel and \mathcal{L} a local logic on its core \mathbf{C} , the *distributed logic* of \mathcal{C} generated by \mathcal{L} , written $DLog_{\mathcal{C}}(\mathcal{L})$, is the inverse image of \mathcal{L} under the sum $f_1 + f_2$.

Refinement: let $\mathcal{C} = \{f_i : \mathbf{A}_i \rightarrow \mathbf{C}\}_{i \in \{1,2\}}$ and $\mathcal{C}' = \{f'_i : \mathbf{A}_i \rightarrow \mathbf{C}'\}_{i \in \{1,2\}}$ be two channels with the same component classifications \mathbf{A}_1 and \mathbf{A}_2 . A *refinement* infomorphism from \mathcal{C}' to \mathcal{C} is an infomorphism $r : \mathbf{C}' \rightarrow \mathbf{C}$ such that for each $i \in \{1,2\}$, $f_i = r \circ f'_i$ (i.e., $\hat{f}_i = \hat{r} \circ \hat{f}'_i$ and $\check{f}_i = \check{f}'_i \circ \check{r}$). Channel \mathcal{C}' is a *refinement* of \mathcal{C} if there exists a refinement infomorphism r from \mathcal{C}' to \mathcal{C} .

B. CHANNEL THEORY THEOREMS

THEOREM B.1. *The logic generated by a classification is sound and complete. Furthermore, given a classification \mathbf{A} and a logic \mathcal{L} on \mathbf{A} , \mathcal{L} is sound and complete if and only if $\mathcal{L} = Log(\mathbf{A})$.*

THEOREM B.2. *Let \mathcal{L} be a logic on a classification \mathbf{B} and $f : \mathbf{A} \rightarrow \mathbf{B}$ an infomorphism.*

1. *If \mathcal{L} is complete then $f^{-1}[\mathcal{L}]$ is complete.*
2. *If \mathcal{L} is sound and \check{f} is surjective then $f^{-1}[\mathcal{L}]$ is sound.*

³All theories considered in this paper are regular.