Seventh International Conference on Principles of Knowledge Representation and Reasoning (KR2000), Breckenridge, Colorado, April 2000.

Ontology-Based Semantics

Mihai Ciocoiu
Dept. of Computer Science
and Institute for Systems Research
University of Maryland
College Park, MD 20742
mihaic@cs.umd.edu

Dana S. Nau
Dept. of Computer Science
and Institute for Systems Research
University of Maryland
College Park, MD 20742
nau@cs.umd.edu

Abstract

We consider the problem of providing semantics for declarative languages, in a way that would be useful for enabling automated knowledge exchange. If we only have one (first order) language, we can formalize what we mean for a set of sentences to be a translation of another, by requiring that the two sets share the same models. However, in order to formalize translation for the case where the two sets of sentences are of different languages, we need a different notion of semantics, capable of overcoming the language barrier. We introduce *Ontology-Based Semantics* with this purpose in mind. We show how ontologies can be used to make implicit assumptions explicit, and how they are integrated in our semantics in order to restrict the set of models a set of sentences has. We show how Ontology-Based Models can be used to formally define knowledge translation for the different language case in a similar way ordinary models can be used to define translation for the one language situation. We also provide a syntactical characterization of knowledge translation, that can be used as an effective procedure to check translatability, and we prove it to be sound and complete with respect to our semantic definition of translation.

1 INTRODUCTION

The ability to translate knowledge from/to different representation languages is an important ingredient for building powerful AI systems, by easing the difficult and time-consuming task of knowledge base construction, and facilitating knowledge sharing among existing ones. In this paper we consider the problem of using formal ontologies for providing semantics to declarative languages, in a way that would be useful for enabling automated knowledge exchange.

Using formal ontologies has been proposed [Gruber 1991] as a solution for managing the inherent heterogeneity present in knowledge from different sources. Different approaches vary in their definition of what a formal ontology is, ranging from taxonomic hierarchies of classes [Campbell et al.], to vocabularies of terms defined by human-readable text, together with sets of formal constraining axioms [Gruber 1993]. Another distinction is the level of commitment of the communicating agents with respect to the shared ontology, varying from having all agents commit to a single common ontology — the standardization approach — to having a network of mediators and facilitators that enable translation among agents' different ontologies [Shave 1997, Gray et al. 1997].

For our purpose we will adopt the logical theory view of an ontology, and the constraining axioms will play a crucial role in defining our semantics. While we allow the communicating agents to have their own declarative languages and ontologies, we will require the existence of a common ontology expressive enough to interpret the concepts in all agents' ontologies. We will also require for a declarative language L that a function σ can be specified that converts sentences of L to sentences of a first order language $\mathcal{L}$. (of course this limits the method's applicability to languages that are not (strictly) more expressive than FOL). Here are some examples of questions we want to address:

- Consider a declarative language L that has a construct like `(mother Bill Anne)`. What would models of this construct look like?
- Consider now the problem of translating sets of sentences among two declarative languages L_1

and L_2. What exactly do we mean when we say that a set S_2 of L_2 sentences is a translation of a set S_1 of L_1 sentences?

For the simplest case, in which we have just one first-order language L and we are considering two sets of L-sentences S_1 and S_2, the obvious solution is to say that S_2 is a *translation* of S_1 if it has the same set of consequences (i.e., $Cn(S_2) = Cn(S_1)$), or equivalently, if S_1 and S_2 share the same set of models (i.e., $\mathfrak{A} \models S_1 \Leftrightarrow \mathfrak{A} \models S_2$ for all $\mathfrak{A}$). One could extend this further by saying that S_2 is a *partial translation* of S_1 if the consequences of S_2 are also consequences of S_1 (i.e., $Cn(S_2) \subseteq Cn(S_1)$), or equivalently, if all models of S_1 are models of S_2 (i.e., $\mathfrak{A} \models S_1 \Rightarrow \mathfrak{A} \models S_2$ for all $\mathfrak{A}$).

Unfortunately, a direct extension of this idea for sets of sentences of two *different* first order languages L_1 and L_2 will not work the way we would like. One problem is that for intuitively similar concepts (and thus ones that we would like translatable) their representations in the two languages might use combinations of functions/predicates of different arities, such as functions in one language and predicates in the other (or any combinations thereof). Thus, models of sets of sentences in the different languages will be different, even if the sets of sentences are intuitively equivalent. For two different arbitrary declarative languages, defining translation is even harder, since we don't even have the notion of a model.

What this paper tries to do is to specify a way for defining models of sets of sentences of arbitrary declarative languages, so that these models can be used to define translation in the same fashion as for the one language situation above.

2 IMPLICIT ASSUMPTIONS

Consider a declarative language L that has a construct like `(mother Bill Anne)`. What would we want models of this construct to look like?

One option would be to follow a database approach: use a closed world assumption, proclaim that we are speaking of a universe with only two persons (`Anne` and `Bill`), and that the motherhood relation holds only for $\langle \texttt{Bill}, \texttt{Anne} \rangle$. In this case, we would have a single model $\mathfrak{A} = (\{\texttt{Bill}, \texttt{Anne}\}, \{\langle \texttt{Bill}, \texttt{Anne} \rangle\})$. However, this approach would not allow us to define partial translation the way we would like, and would prohibit translating bits and pieces of information from more expressive languages into less expressive ones. Instead, we would prefer the semantics of `(mother Bill Anne)` to be "the universe includes `Anne` and `Bill` and maybe other persons and the motherhood relation holds for $\langle \texttt{Bill}, \texttt{Anne} \rangle$ and maybe for some other pairs," like the semantics for FOL is usually defined.

The first temptation would be to define the relation σ so that:

1. σ maps an L construct such as `(mother ?x ?y)` into a predicate such as Mother(x, y) of the first order language $\mathcal{L}$;
2. the models of a set of L sentences (e.g., $S_1 = \{$`(mother Bill Anne)`$\}$) are in fact the models of the set $\Sigma_1 = \{\sigma(s) : s \in S_1\}$; (e.g., the models of {Mother(Bill, Anne)}).

This would allow other wanted structures, that include different persons, to be considered, like $\mathfrak{B} = (\{\texttt{Bill}, \texttt{Anne}, \texttt{Cathy}\}, \{\langle \texttt{Bill}, \texttt{Anne} \rangle, \langle \texttt{Anne}, \texttt{Cathy} \rangle\})$. However, it would also allow unwanted models to creep in, such as:

$\mathfrak{C} = (\{\texttt{Bill}, \texttt{Anne}\}, \{\langle \texttt{Bill}, \texttt{Anne} \rangle, \langle \texttt{Anne}, \texttt{Anne} \rangle\})$.

$\mathfrak{D} = (\{\texttt{Bill}, \texttt{Anne}, \texttt{Cathy}\}, \{\langle \texttt{Bill}, \texttt{Anne} \rangle, \langle \texttt{Bill}, \texttt{Cathy} \rangle\})$.

$\mathfrak{E} = (\{\texttt{Bill}, \texttt{Anne}, \texttt{Cathy}\}, \{\langle \texttt{Bill}, \texttt{Anne} \rangle, \langle \texttt{Anne}, \texttt{Cathy} \rangle, \langle \texttt{Cathy}, \texttt{Bill} \rangle\})$.

The problem is that there are implicit assumptions in the language L, such as the fact that a person cannot be her own mother, that one cannot have two different mothers, etc. These assumptions need to be made explicit in order to define the correct set of models for `(mother Bill Anne)`.

Making such assumptions explicit has become known in AI as building the domain ontology [Gruber 1991], and it holds the promise of enabling knowledge exchange.

The above considerations lead us to explicitly constructing a motherhood ontology Ω into $\mathcal{L}$ and taking the models of the L sentence `(mother Bill Anne)` to be the models of the $\mathcal{L}$ theory having as axioms the sentence's image through σ together with the ontology itself, i.e. the models of $\Omega \cup \{$Mother(Bill, Anne)$\}$. Will this do?

Well, it will give us the models we wanted, but will not help much with translation. If some other language defines in its ontology a motherhood relation, then conceivably an automated translation procedure would identify the motherhood predicates as mutually translatable, but it would also translate to motherhood all predicates that satisfy the motherhood ontology. For example, the successor predicate that holds between an integer number and its successor satisfies all of the motherhood axioms, and so an automated translation procedure will consider a (partial) translation of `(s ?x ?y)` to `(mother ?x ?y)`. (The translation will only be one-way, since the successor relation satisfies additional constraints, being in fact a bijective function, as opposed to the motherhood, which is only surjective.)

A much better idea is to *share* the motherhood ontology. Sharing ontologies will greatly simplify the task of a semantic-based automated translation procedure and also have the additional benefit of simplifying the process of writing ontologies, by enabling the reuse of already-existing components. In the next section we will present a way to define semantics that makes use of the ontology sharing idea.

3 ONTOLOGY-BASED SEMANTICS

3.1 LOGICAL RENDER

As we mentioned earlier, we are interested in providing semantics for *declarative languages*, which for our purpose are languages L such that a function σ can be specified that converts sentences of L to sentences of a first order language $\mathcal{L}$.

We call such a function σ a *logical rendering* function, and the image Σ of a set S of L sentences through σ the *logical render* of S through σ.

Coming back to our motherhood example, the *logical rendering* function σ will convert instances of `(mother ?x ?y)` to corresponding instances of the $\mathcal{L}$ predicate Mother(x, y).

3.2 INTERPRETATIONS

To simplify our definitions we will restrict ourselves to first order languages $\mathcal{L}$ that contain no function symbols. Note that this is not a reduction of expressivity, since any formula of a first order language $\mathcal{L}$ that includes function symbols can be converted to a formula of a language $\mathcal{L}$' similar to $\mathcal{L}$ but which has no function symbols and has additional $(n+1)$-ary predicates corresponding to each n-ary function of $\mathcal{L}$.

Our notion of interpretation is the restriction to predicate calculus of the standard mathematical one, as it appears in [Enderton 1972].

Definition An *interpretation* π of a function-free language $\mathcal{L}$ into a theory T of language $\mathcal{L}_\Omega$ is a function on the set of parameters of $\mathcal{L}$ such that:

1. π assigns to $\forall$ a formula $\pi_\forall$ of $\mathcal{L}_\Omega$ in which at most one variable v_1 occurs free, such that $T \models \exists v_1 \pi_\forall$.
2. π assigns to each n-place predicate symbol P a formula π_P of $\mathcal{L}_\Omega$ in which at most n variables $v_1, \ldots, v_n$ occur free.

Definition An *interpretation* φ^π of a $\mathcal{L}$-formula φ is recursively defined in the obvious way: i.e. if φ is an atomic formula P, its interpretation is the formula π_P applied to the same set of constants/variables; otherwise $(\neg\varphi)^\pi$ is $(\neg\varphi^\pi)$, $(\varphi \rightarrow \psi)^\pi$ is $(\varphi^\pi \rightarrow \psi^\pi)$, $(\forall x \varphi)^\pi$ is $\forall x(\pi_\forall(x) \rightarrow \varphi^\pi)$, etc.

Definition An *interpretation* π of a theory T_0 of language $\mathcal{L}$ into a theory T of language $\mathcal{L}_\Omega$ is an interpretation π of the language $\mathcal{L}$ into T such that for all $\mathcal{L}$-sentences φ, $\varphi \in T_0 \Rightarrow \varphi^\pi \in T$.

3.3 EXPLANATIONS

Definition Given a domain ontology Ω expressed as a set of $\mathcal{L}_\Omega$ sentences, a theory T of $\mathcal{L}_\Omega$ is called a *domain theory* for Ω iff $T \models \Omega$.

Definition Given an interpretation π, a logical rendering function σ and an ontology Ω (expressed as a set of $\mathcal{L}_\Omega$ sentences), an $\mathcal{L}_\Omega$ *explanation* of a set S of L sentences is a $\mathcal{L}_\Omega$ domain theory T for Ω, such that if Σ is the logical render of S through σ, $T \models \Sigma^\pi$.

Intuitively, an $\mathcal{L}_\Omega$ *explanation* of a set S of sentences of language L is a theory of $\mathcal{L}_\Omega$ that has among its axioms the interpretation of the rendering of S, with the concepts that appear in them "explained" by ontology Ω. Going back to our example, an explanation of `(mother Bill Anne)` is a theory that has Mother(Bill, Anne) and the motherhood ontology as axioms.

3.4 ONTOLOGY-BASED MODELS

Given a model $\mathfrak{A}$ of an explanation T of a set of L sentences S, we can extract from it a model ${}^\pi\mathfrak{A}$ of the render Σ of S, that has the desired property of obeying the additional constraints imposed by the ontology Ω. Namely, let

1. $|{}^\pi\mathfrak{A}|$ = the set defined in $\mathfrak{A}$ by $\pi_\forall$;
2. $P^{{}^\pi\mathfrak{A}}$ = the relation defined in $\mathfrak{A}$ by π_P, restricted to $|{}^\pi\mathfrak{A}|$.

${}^\pi\mathfrak{A}$ is called an Ontology-Based Model of S (written ${}^\pi\mathfrak{A} \models^\Omega_{\sigma,\pi} S$).

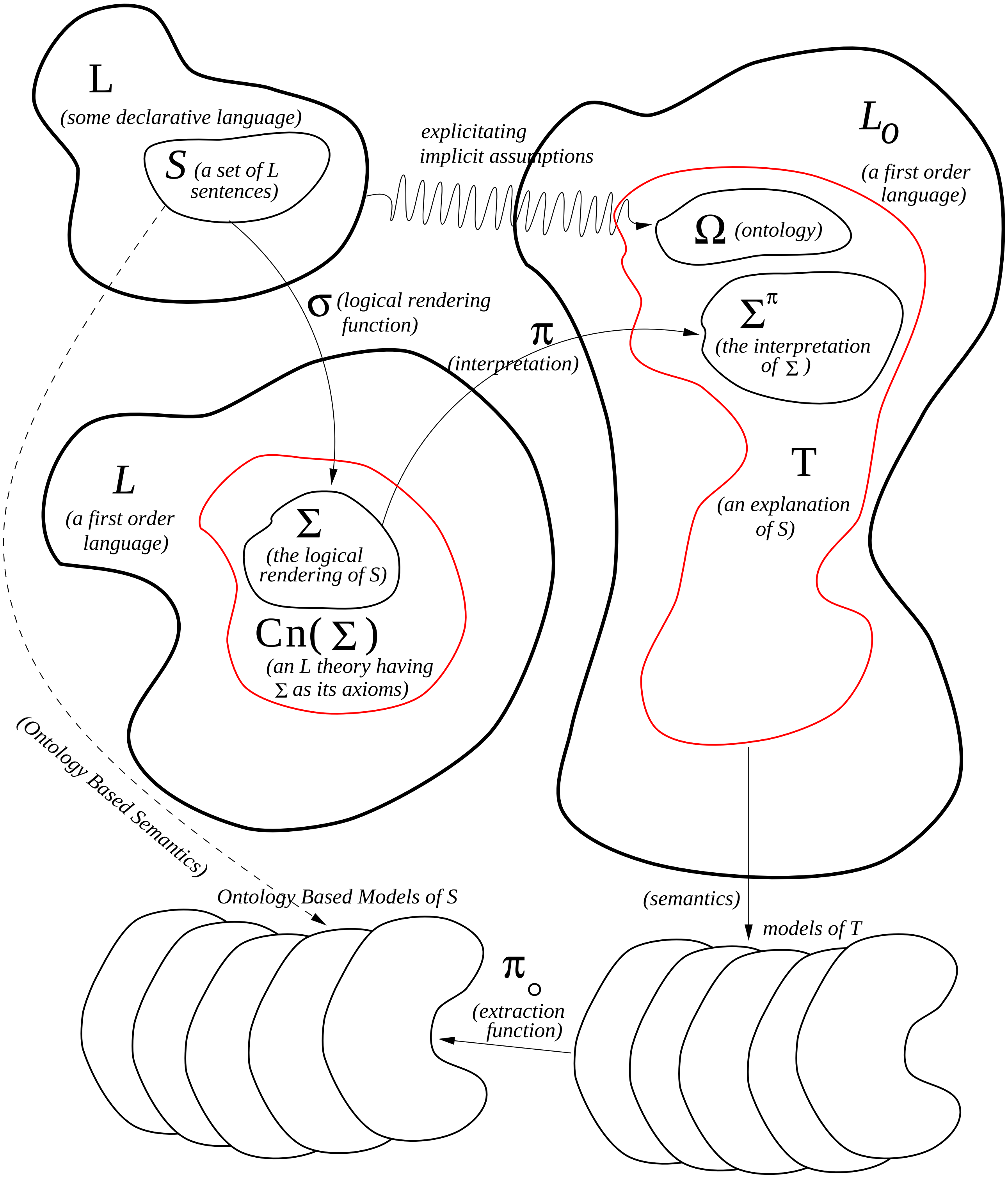

Figure 1: Ontology-Based Models

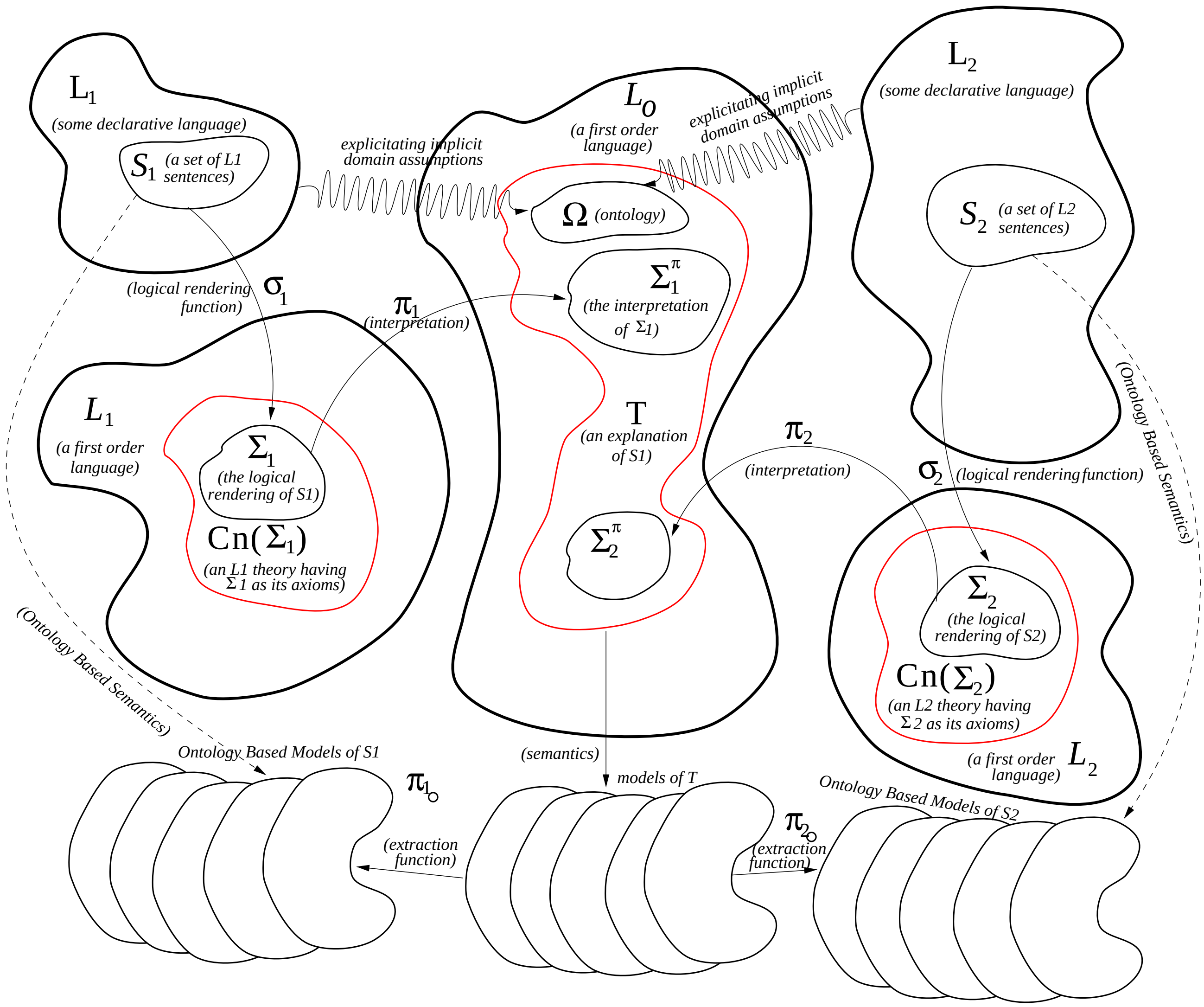

Figure 2: Ontology-Based Translation

4 ONTOLOGY-BASED TRANSLATION

Ontology-Based Models allow a definition of translation for the different language situation in the same way that ordinary models allowed it for the single-language case.

Suppose we are given two declarative languages L_1 and L_2, a domain ontology Ω (expressed as a set of sentences in language $\mathcal{L}_\Omega$), rendering functions σ_1 and σ_2, and interpretations π_1 and π_2. Then a set S_2 of L_2 sentences is an *Ontology-Based Partial Translation* of a set S_1 of L_1 sentences iff for every model $\mathfrak{A}$ of every explanation T of S_1, ${}^{\pi_1}\mathfrak{A}\models^{\Omega}_{\sigma_1,\pi_1} S_1 \Rightarrow {}^{\pi_2}\mathfrak{A}\models^{\Omega}_{\sigma_2,\pi_2} S_2$.

Similarly, suppose we are given two declarative languages L_1 and L_2, a domain ontology Ω (expressed as a set of sentences in language $\mathcal{L}_\Omega$), rendering functions σ_1 and σ_2, and interpretations π_1 and π_2. Then a set S_2 of L_2 sentences is an *Ontology-Based Translation* of a set S_1 of L_1 sentences iff S_2 is an Ontology Based Partial Translation of S_1 and S_1 is an Ontology-Based Partial Translation of S_2.

5 AN EXAMPLE TRANSLATION

Suppose we have a declarative language L_1 that has a construct like `(GM ?x ?y)` whose intended semantics is “`?y` is an grandmother of `?x`”, and another declarative language L_2 that has a construct `Y anc X` whose intended semantics is “`Y` is an ancestor of `X`”. In order to provide ontology-based semantics for those two languages, we must first build a domain ontology Ω (in our case a family ontology), and provide the logical rendering (σ_1 and σ_2) and interpretation functions (π_1 and π_2).

5.1 A FAMILY ONTOLOGY

As primitive concepts, our toy family ontology will have the concepts of $Male$ and $Female$. As a primitive relation, it will have the parenthood relation $Parent(x, y)$, which holds if y is a parent of x. As a defined relation it will introduce $Ancestor(x, y)$ by the following two axioms:

$$(\forall x)(\forall y)\ Parent(x, y) \rightarrow Ancestor(x, y)$$

$$(\forall x)(\forall y)(\forall z)\ Parent(x, z) \wedge Ancestor(z, y) \rightarrow Ancestor(x, y)$$

Suppose we also write the following axioms in order to constrain the possible interpretation of the primitive concepts:

$(\forall x)\ \neg(Male(x) \wedge Female(x))$; i.e., $Male$ and $Female$ are disjoint concepts.

$(\forall x)\ \neg Parent(x, x)$; i.e., One cannot be his/hers own parent.

$(\forall x)(\forall y)\ \neg(Parent(x, y) \wedge Ancestor(y, x))$; i.e., One cannot be a parent of one of his/hers ancestors.

$(\forall x)(\forall y)(\forall z)\ (Parent(x, y) \wedge Parent(x, z) \wedge Female(y) \wedge Female(z)) \rightarrow y = z$; i.e., One's $Female$ parent is unique.

$(\forall x)(\forall y)(\forall z)\ (Parent(x, y) \wedge Parent(x, z) \wedge Male(y) \wedge Male(z)) \rightarrow y = z$; i.e., One's $Male$ parent is unique.

5.2 LOGICAL RENDERING AND INTERPRETATION FUNCTIONS

The logical rendering functions for this simple example would just convert the given constructs into the corresponding first-order atomic sentences, i.e. the logical render of `(GM ?x ?y)` would be the $\mathcal{L}_1$ atomic sentence $GM(x, y)$ and the logical render of `Y anc X` would be the $\mathcal{L}_2$ atomic sentence $Anc(x, y)$.

The interpretation of $GM(x, y)$ will be a $\mathcal{L}_\Omega$ formula having at most two free variables, in our case $(\exists z)\ Parent(x, z) \wedge Parent(z, y) \wedge Female(y)$.

Finally, the interpretation of $Anc(x, y)$ must be a $\mathcal{L}_\Omega$ formula having at most two free variables, in our case $Ancestor(x, y)$.

5.3 EXAMPLE TRANSLATION

In this section we show how the syntactic characterization of Ontology-Based Translation can be used to verify translatability for a given set of ground sentences of two different languages.

Consider the L_1 theory S_1= {`(GM Bill, Anne), (GM Anne, Cathy)`}, and the L_2 theory S_2= {`Anne anc Bill, Cathy anc Anne, Cathy anc Bill`}. Then S_2 is an ontology-based partial translation of S_1, but S_1 is *not* an ontology-based partial translation of S_2. Indeed, the $\mathcal{L}_1$ and $\mathcal{L}_2$ logical renders of S_1 and S_2 are:

$$\Sigma_1 = \{GM(Bill, Anne),\ GM(Anne, Cathy)\} \text{ and}$$
$$\Sigma_2 = \{Anc(Bill, Anne),\ Anc(Anne, Cathy),\ Anc(Bill, Cathy)\}$$

and their $\mathcal{L}_\Omega$ interpretations are:

$$\Sigma_1^{\pi_1} = \{(\exists z)\ Parent(Bill, z) \wedge Parent(z, Anne) \wedge Female(Anne),\ (\exists z)\ Parent(Anne, z) \wedge Parent(z, Cathy) \wedge Female(Cathy)\} \text{ and}$$
$$\Sigma_2^{\pi_2} = \{Ancestor(Bill, Anne),\ Ancestor(Anne, Cathy),\ Ancestor(Bill, Cathy)\}$$

It is easy to show that $(\Sigma_1^{\pi_1} \cup \Omega) \vdash \Sigma_2^{\pi_2}$, and this is a necessary and sufficient condition (see Section 6 for a proof) for S_2 to be an ontology-based partial translation of S_1. However, the reverse is not true. S_1 is not an ontology-based partial translation of S_2. To see this, consider an explanation T of S_2,

$$T = Cn(\Omega \cup \{Parent(Bill, Anne),\ Parent(Anne, Cathy)\})$$

and a model of it:

$$\mathfrak{A} = (\{Anne, Bill, Cathy\},\ Parent^{\mathfrak{A}} = \{\langle Bill, Anne\rangle, \langle Anne, Cathy\rangle\},\ Male^{\mathfrak{A}} = \{Bill\},\ Female^{\mathfrak{A}} = \{Anne, Cathy\},\ Ancestor^{\mathfrak{A}} = \{\langle Bill, Anne\rangle, \langle Anne, Cathy\rangle, \langle Bill, Cathy\rangle\}).$$

Note that ${}^{\pi_2}\mathfrak{A} \models^{\Omega}_{\sigma_2,\pi_2} S_2$, but ${}^{\pi_1}\mathfrak{A} \not\models^{\Omega}_{\sigma_1,\pi_1} S_1$, so S_1 is not an ontology-based partial translation of S_2. Also note that neither $S_1' =$ {`(GM Bill Cathy)`} nor any other non-void set S_1 of L_1 sentences could be an ontology-based partial translation of S_2, since we can always construct models $\mathfrak{A}$ of S_2 explanations such that ${}^{\pi_1}\mathfrak{A}$ are not ontology-based models of S_1, by violating the sex/parenthood constraints necessary for the grandmotherhood relation of L_1. (Such a class of models would be the ones which interpret the motherhood relation by the void set.)

In practice, an automated inference procedure could be used on the domain ontology and the logical rendering and interpretation functions, in order to precompile a generic rule of the form "`?y anc ?x` is an ontology-based translation of `(GM ?x ?y)`", and then use it for generating efficient direct translators among the two languages.

6 A SYNTACTIC CHARACTERIZATION

Theorem 6.1 *(Soundness and Completeness of Syntactic Characterization)*

A set S_2 of L_2-sentences is an ontology-based partial translation *of a set S_1 of L_1-sentences (with respect to rendering functions σ_1 and σ_2 and interpretations π_1 and π_2) iff*

$$(\Sigma_1^{\pi_1} \cup \Omega) \vdash \Sigma_2^{\pi_2}$$

where Σ_1 and Σ_2 are the logical renders of S_1 and S_2 through σ_1 and σ_2.

Lemma 6.2 *If ${}^{\pi}\mathfrak{A}$ is an ontology-based model of a set of sentences S with respect to logical rendering function σ and interpretation π, ${}^{\pi}\mathfrak{A} \models^{\Omega}_{\sigma,\pi} S$, then $\mathfrak{A}$ is a model of the interpretation Σ^{π} of the logical render Σ of S, i.e. $\mathfrak{A} \models \Sigma^{\pi}$.*

Proof (Lemma 6.2) Suppose $\mathfrak{A} \not\models \Sigma^{\pi}$. By definition of an ontology-based model, there must exist an $\mathcal{L}_{\Omega}$ explanation T of S, such that $T \models \Omega$ and $T \models \Sigma^{\pi}$, and a model $\mathfrak{B}$ of T such that ${}^{\pi}\mathfrak{B} = {}^{\pi}\mathfrak{A}$. (If such a model doesn't exist, then ${}^{\pi}\mathfrak{A}$ cannot be an ontology-based model of S.)

Since T is a theory and $T \models \Sigma^{\pi}$, it must be the case that $\Sigma^{\pi} \subseteq T$. Since $\mathfrak{B} \models T$, $\mathfrak{B}$ is a model for Σ^{π}, $\mathfrak{B} \models \Sigma^{\pi}$. Since $\mathfrak{A} \not\models \Sigma^{\pi}$, there must exist a sentence $\gamma^{\pi} \in \Sigma^{\pi}$ such that $\mathfrak{A} \not\models \gamma^{\pi}$. It can be proved by induction on the structure of γ that this cannot be the case.

Proof (Theorem 6.1, soundness) Suppose $(\Sigma_1^{\pi_1} \cup \Omega) \vdash \Sigma_2^{\pi_2}$. Consider an arbitrary domain theory T, and an arbitrary model $\mathfrak{A}$ of T. If ${}^{\pi_1}\mathfrak{A} \models^{\Omega}_{\sigma_1,\pi_1} S_1$, then by Lemma 6.2, $\mathfrak{A} \models \Sigma_1^{\pi_1}$. Since T is a domain theory, $\Omega \subseteq T$; and since $\mathfrak{A}$ is a model of T, $\mathfrak{A} \models \Omega$; and thus $\mathfrak{A} \models (\Sigma_1^{\pi_1} \cup \Omega)$. Since T is a theory, then by our supposition that $(\Sigma_1^{\pi_1} \cup \Omega) \vdash \Sigma_2^{\pi_2}$, it follows that $\Sigma_2^{\pi_2} \subseteq T$, and thus T is an explanation of S_2.

Since $\mathfrak{A}$ is a model of T, ${}^{\pi_2}\mathfrak{A}$ is an ontology-based model for S_2 i.e., ${}^{\pi_2}\mathfrak{A} \models^{\Omega}_{\sigma_2,\pi_2} S_2$. Since T and $\mathfrak{A}$ were arbitrarily chosen, it follows that for every model $\mathfrak{A}$ of every domain theory T, ${}^{\pi_1}\mathfrak{A} \models^{\Omega}_{\sigma_1,\pi_1} S_1 \Rightarrow {}^{\pi_2}\mathfrak{A} \models^{\Omega}_{\sigma_2,\pi_2} S_2$, and thus S_2 is an ontology-based partial translation of S_1.

Proof (Theorem 6.1, completeness) Suppose S_2 is an ontology-based partial translation of S_1. Consider an arbitrary model $\mathfrak{A} \models (\Sigma_1^{\pi_1} \cup \Omega)$ and let $T = Cn(\Sigma_1^{\pi_1} \cup \Omega)$. T is an explanation of S_1; and since $\mathfrak{A} \models (\Sigma_1^{\pi_1} \cup \Omega)$, $\mathfrak{A} \models T$.

By our supposition and the definition of ontology-based translation, it follows that ${}^{\pi_2}\mathfrak{A} \models^{\Omega}_{\sigma_2,\pi_2} S_2$. By Lemma 6.2 it follows that $\mathfrak{A} \models \Sigma_2^{\pi_2}$. Since $\mathfrak{A}$ was arbitrary chosen, $(\Sigma_1^{\pi_1} \cup \Omega) \models \Sigma_2^{\pi_2})$; and by completeness of first order deduction, $(\Sigma_1^{\pi_1} \cup \Omega) \vdash \Sigma_2^{\pi_2}$.

7 CONCLUSIONS

In this paper we have presented a new kind of semantics, called *Ontology-Based Semantics*, intended for facilitating automated knowledge exchange between declarative languages. Our results include the following:

We have shown how domain specific information, encoded as ontologies, is used in constructing *Ontology-Based Models* that restrict the possible interpretations a set of sentences can have.

We have shown how Ontology-Based Models can be used to formally define knowledge translation for the different-language case, in a way similar to how ordinary models can be used to define translation for the one-language situation.

We have provided a syntactical characterization of knowledge translation, that can be used as an effective procedure to check translatability, and we have proved it to be sound and complete with respect to our semantic definition of translation.

The principal benefit of our semantics is that it provides a formal foundation for reasoning about the properties of systems that do automated knowledge translation based on ontology sharing. As part of our collaboration with NIST on their Process Specification Language (PSL) project [Schlenoff et al. 1999, Schlenoff et al. 1999], we plan to develop a system that would be able to automatically generate efficient translators based on declarative languages' ontology-based semantics, specified as a logical rendering function and a PSL interpretation.

Acknowledgements

This work was supported in part by the following grants and contracts: National Institute for Standards and Technology `70NANB6H0147`, Army Research Laboratory `DAAL01-97-K0135`, Air Force Research Laboratory `F306029910013`, and National Science Foundation `DMI-9713718`.

References

[Campbell et al.] Alistair E. Campbell, Hans Chalupsky, and Stuart C. Shapiro. Ontological Mediation: An Analysis. Unpublished manuscript.

[Enderton 1972] H. B. Enderton. *A Mathematical Introduction to Logic.* Academic Press, 1972.

[Gray et al. 1997] P. M. D. Gray et al. KRAFT: Knowledge Fusion from Distributed Databases and Knowledge Bases. In *The 8th International Conference and Workshop on Database and Expert Systems Applications*, Toulouse, France, September 1997.

[Gruber 1991] Thomas R. Gruber. The Role of Common ontology in achieving sharable, reusable knowledge bases. In Richard Fikes, James A. Allen, and Erik Sandewall, editors, *Proceedings of the Second International Conference*, Principles of Knowledge Representation and Reasoning, 1991.

[Gruber 1993] Thomas R. Gruber. Toward Principles of the Design of Ontologies Used for Knowledge Sharing. In *International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation*, Padova, Italy, 1993.

[Schlenoff et al. 1999] Craig Schlenoff, Mihai Ciocoiu, Don Libes, and Michael Gruninger. Process Specification Language: Results of the First Pilot Implementation In *Proceedings of the 1999 International Mechanical Engineering Congress and Exposition (IMECE)*, November 1999.

[Schlenoff et al. 1999] Craig Schlenoff, Michael Gruninger, and Mihai Ciocoiu. The Essence of Process Specification Submitted, by invitation, to the *Special Issue on Modeling and Simulation of Manufacturing Systems for the Society for Computer Simulation International*, 1999.

[Shave 1997] M. J. R. Shave. Ontological Structures for Knowledge Sharing In *New Review of Information Networking*, 1997.