A Foundation for a Semantic Web

John D. Ramsdell*
The MITRE Corporation

October 13, 2000

Abstract

A semantic web is a web of data designed to be processed by machines. It enables processing based on the meaning of the data. To be useful, semantic data will be combined from several sources. This paper focuses on the relation of the combined data to its sources. Using the method of interpretations between theories in a logic with undefined terms, it establishes criteria for combining information in a fashion that preserves the inferences available in the original information. The formalism can be used to evaluate existing languages for semantic data on the web, such as the Simple HTML Ontology Extensions (SHOE).

1 Logic and the Web

In the traditional web, information is structured and shared in forms that facilitate its display for human consumption. For example, an HTML document is divided into sections, paragraphs, and lists. The document's structure guides a browser's rendering of the document on a computer screen.

*Supported by the MITRE Sponsored Research Program. Author's address: The MITRE Corpopation, 202 Burlington Road, Bedford, MA 01730.

The structure offers little assistance to programs extracting information from the web. This is because the structure of a document usually has little to do with the meaning of its contents. As a result, general purpose methods for extracting information rely on other metrics to associate meaning with a document.

For example, search engines categorize web pages by looking at word occurrences. A document that contains the word "genealogy" is assumed to be about genealogy. I might search the web for documents that contain the words "Ramsdell" and "genealogy" to find my ancestors with my surname. Such a search would return this document, a document about logic and the web, not genealogy. Still, the search is likely to point me to a site that contains information about ancestors with my surname.

Suppose instead that I would like find the names of all of my ancestors that were ministers. Since this search involves following both my maternal as well as paternal blood lines, one must expect to visit many different sites during the search. Each site will have its own unique structure, a structure guided by its author's opinion on how to break each web page into sections, paragraphs, and lists, not by its genealogy content. In the traditional web, writing a program to extract this kind of information is nearly impossible.

In [1], Tim Berners-Lee promotes the creation of what he calls a *Semantic Web*—a web of data designed to be processed by machines. The vision is that each participating site would augment its material with a machine-oriented version of its data. Continuing the genealogy example, site A might publish machine-oriented data using the following schema:

$$\begin{array}{rll} \mathsf{parent\_of}(x,y) & \text{means} & \text{person } x \text{ is the parent of person } y, \\ \mathsf{employed\_as}(x,y) & \text{means} & \text{person } x \text{ was employed as a } y, \\ \mathsf{is\_male}(x) & \text{means} & \text{person } x \text{ is a male,} \\ \mathsf{is\_female}(x) & \text{means} & \text{person } x \text{ is a female.} \end{array}$$

The gender predicates are used later in the section on interpretations between theories.

Figure 1 gives a machine-oriented semantics for data in the above schema. In particular, it defines the concept of an ancestor.

If every genealogy site used the same schema and agreed on the terms used to identify people and jobs, finding ancestors that are ministers would be easy. One would simply use the obvious inference rule to derive ancestors

$$\begin{array}{rcl}
\mathsf{parent_of}(x,y) & \text{implies} & \mathsf{is_person}(x) \\
\mathsf{parent_of}(x,y) & \text{implies} & \mathsf{is_person}(y) \\
\mathsf{employed_as}(x,y) & \text{implies} & \mathsf{is_person}(x) \\
\mathsf{employed_as}(x,y) & \text{implies} & \mathsf{is_job}(y) \\
\mathsf{is_male}(x) & \text{implies} & \mathsf{is_person}(x) \\
\mathsf{is_female}(x) & \text{implies} & \mathsf{is_person}(x) \\
\mathsf{parent_of}(x,y) & \text{implies} & \mathsf{ancestor}(x,y) \\
\mathsf{parent_of}(x,y) \text{ and } \mathsf{ancestor}(y,z) & \text{implies} & \mathsf{ancestor}(x,z)
\end{array}$$

Figure 1: Schema Semantics

from the parent of relation in combination with the data from the genealogy sites to answer the question.

It is unrealistic to assume the existence of a common schema on the web. The web is constantly changing. Sites are developed independently, and one must be able to combine the information from sites in new ways without modifying existing sites. Furthermore, the information must be combined in a fashion that preserves the inferences available in the original information.

This paper proposes the use of well established methods in Mathematical Logic to precisely define the essential properties of information combined from a semantic web. A variation of First-Order Logic called Partial First-Order Logic (PFOL) will form the foundation of the definition. Interpretations between theories contributes the essential properties of information combination.

The formalism can be used to evaluate existing languages used to provide machine-oriented data on the web, and suggest improvements to the language. The Simple HTML Ontology Extensions (SHOE) [4] is an ontology-based knowledge representation language that is embedded in web pages. The authors provide a mapping of SHOE into First-Order Logic [5], so it is trivial to evaluate it within the formalism. Interpretations between theories may suggest more expressive ways of combining SHOE specifications.

The formalism could also be used as the basis of a language for providing machine-oriented data, however, this possibility will not be pursued in this paper.

2 Partial First-Order Logic

PFOL is very similar to First-Order Logic; the difference is that terms in PFOL may be undefined, and atomic formulas that contain undefined terms are false.

The availability of undefined terms provides several advantages. Many specifications are more naturally described in the presence of undefined terms, however, the relevant reason this formalism is built on PFOL is it allows interpretations between theories in which each term is translated into another term. In First-Order Logic, an atomic formula that contains the application of an operator symbol to a sequence of terms is translated into a complex formula because the translation of the application may be undefined in some domains.

The variant of Partial First-Order Logic used here is taken from [3] where it forms the basis of a set theory with support for partial functions. The paper also describes a sort system used to classify terms. PFOL, the set theory, and the sort system are intended to serve as a foundation for mechanized mathematical systems. The set theory, and the sort system may find use in a semantic web.

2.1 Syntax

A *variable* of PFOL is a member of a fixed infinite set $\mathcal{V}$ of symbols. A *language* of PFOL is a tuple $(\mathcal{C}, \mathcal{O}, \mathcal{P})$ such that:

1. $\mathcal{C}$ is a set of *individual constants.*

2. $\mathcal{O}$ is a set of *operator symbols,* each with an assigned arity ≥ 1.

3. $\mathcal{P}$ is a set of *predicate symbols,* each with an assigned arity ≥ 1. $\mathcal{P}$ contains the binary predicate symbol $=$.

4. $\mathcal{V}$, $\mathcal{C}$, $\mathcal{O}$, and $\mathcal{P}$ are pointwise disjoint.

A language in which $\mathcal{O}$ is empty is called *operator-free.* In the remainder of this section, let $\mathcal{L} = (\mathcal{C}, \mathcal{O}, \mathcal{P})$ be a language of PFOL.

A *term* and a *formula* of $\mathcal{L}$ are defined inductively by:

1. Each $x \in \mathcal{V}$ and $a \in \mathcal{C}$ is a term.

2. If $x \in \mathcal{V}$ and ψ is a formula, then $(\mathrm{I}x.\, \psi)$ is a term.

3. If $o \in \mathcal{O}$ is n-ary and $t_1, \ldots, t_n$ are terms, then $o(t_1, \ldots, t_n)$ is a term.

4. If $p \in \mathcal{P}$ is n-ary and $t_1, \ldots, t_n$ are terms, then $p(t_1, \ldots, t_n)$ is a formula.

5. If φ and ψ are formulas and $x \in \mathcal{V}$, then $\neg\varphi$, $(\varphi \rightarrow \psi)$, and $(\forall x. \varphi)$ are formulas.

The symbols $\neg$, $\rightarrow$ $\forall$, and I are the *logical constants* of PFOL. The symbols are operators for negation, implication, universal quantification, and definite description. No logical constant is a member of any of the sets $\mathcal{V}$, $\mathcal{C}$, $\mathcal{O}$, or $\mathcal{P}$. A term or formula is *regular* if it does not contain any occurrences of the definite description operator. Regular terms and formulas are the same as the terms and formulas of ordinary first-order logic.

Parentheses in terms and formulas may be suppressed when meaning is not lost. For convenience, we also employ the following abbreviations:

$$\begin{array}{rll} (t_1 = t_2) & \text{for} & = (t_1, t_2), \\ (\varphi \wedge \psi) & \text{for} & \neg(\varphi \rightarrow \neg\psi), \\ (\varphi \vee \psi) & \text{for} & \neg\varphi \rightarrow \psi, \\ (\exists x. \varphi) & \text{for} & \neg\forall x. \neg\psi. \end{array}$$

Let an *expression* of $\mathcal{L}$ be either a term or a formula of $\mathcal{L}$. "Free variable", "closed", and similar notions are defined in the obvious way. A *sentence* is a closed formula. Given an a formula φ, $\varphi[x \mapsto t]$ is the result of simultaneously replacing each free occurrence of the variable x in φ with term t. A set of formulas Γ is said to be *over* $\mathcal{L}$ iff each $\varphi \in \Gamma$ is a formula of $\mathcal{L}$.

Returning to the genealogy example, site B might publish machine-oriented data using the following schema:

$$\begin{array}{rll} x = \mathsf{dad}(y) & \text{means} & \text{person } x \text{ is the father of person } y, \\ x = \mathsf{mom}(y) & \text{means} & \text{person } x \text{ is the mother of person } y, \\ \mathsf{job}(x, y) & \text{means} & \text{person } x \text{ was employed as a } y. \end{array}$$

The term $\mathsf{dad}(y)$ is defined when y is a person. The use of the dad operator symbol embeds the assumptions that one has exactly one father. In this schema, the gender of a person can only be determined if that person is a parent.

2.2 Semantics

Let $\underline{\mathcal{L}} = (\underline{\mathcal{C}}, \emptyset, \underline{\mathcal{P}})$ be an operator-free language. PFOL and First-Order Logic share the same notion of truth and models for any set of regular PFOL formulas over $\underline{\mathcal{L}}$. A *structure* $\mathcal{U}$ associates a nonempty set $\mathcal{U}_\forall$ with the universal quantifier, an element $\mathcal{U}_a \in \mathcal{U}_\forall$ with each constant symbol $a \in \underline{\mathcal{C}}$, and an n-ary relation $\mathcal{U}_p^n \subseteq \mathcal{U}_\forall^n$ with each n-ary predicate symbol $p \in \underline{\mathcal{P}}$. $\mathcal{U}_=$ is always the identity relation

$$\mathcal{U}_= = \{(u, u) : u \in \mathcal{U}_\forall\}.$$

A *variable assignment* into $\mathcal{U}$ is a function which maps each $x \in \mathcal{V}$ into an element of $\mathcal{U}_\forall$. Given a variable assignment A into $\mathcal{U}$, $x \in \mathcal{V}$, and $u \in \mathcal{U}_\forall$, let $A[x \mapsto u]$ be the variable assignment A' into $\mathcal{U}$ such that $A'(x) = u$ and $A'(y) = A(y)$ when $y \neq x$.

An *extended variable assignment* into $\mathcal{U}$ is a function which maps each $z \in \mathcal{V} \cup \mathcal{C}$ into an element of $\mathcal{U}_\forall$. Given a variable assignment A into $\mathcal{U}$, the extended variable assignment variable assignment $\overline{A}$ is given by

$$\overline{A}(z) = \begin{cases} A(z) & z \in \mathcal{V} \\ \mathcal{U}_z & z \in \mathcal{C} \end{cases}$$

Given a regular formula φ over an operator-free language, $\mathcal{U}$ *satisfies* φ *with* A, when φ meets the conditions given by the matching formula:

1. $\mathcal{U}$ satisfies $p(t_1, \ldots t_n)$ with A, iff $(\overline{A}(t_1), \ldots, \overline{A}(t_n)) \in \mathcal{U}_p$.

2. $\mathcal{U}$ satisfies $\neg\varphi$ with A, iff $\mathcal{U}$ does not satisfy φ with A.

3. $\mathcal{U}$ satisfies $\varphi \to \psi$ with A, iff $\mathcal{U}$ does not satisfy φ with A, or $\mathcal{U}$ satisfies ψ with A.

4. $\mathcal{U}$ satisfies $\forall x.\, \varphi$ with A, iff $\mathcal{U}$ satisfies φ with $A[x \mapsto u]$ for all $u \in \mathcal{U}_\forall$.

A set of formulas Γ *logically implies* a formula φ, written $\Gamma \models \varphi$, if for every structure $\mathcal{U}$ of the language and every variable assignment A such that $\mathcal{U}$ satisfies every member of Γ with A, then $\mathcal{U}$ satisfies φ with A.

Structure $\mathcal{U}$ is a *model* of φ iff $\mathcal{U}$ satisfies φ with A for all variable assignments A. Structure $\mathcal{U}$ is a *model* of a set of formulas Γ iff it is a model of every member of Γ.

2.2.1 Semantics of Full PFOL

Structures for First-Order Logic associate a total function with each operator symbol, while structures for PFOL associate a partial function with each operator symbol. The definition of a model for PFOL is given in [3]; this paper will give the semantics of PFOL by translating arbitrary formulas into regular formulas over an operator-free language.

Assume $\mathcal{L} = (\mathcal{C}, \mathcal{O}, \mathcal{P})$ and define $\underline{\mathcal{P}} = \mathcal{P} \cup \{p_o : o \in \mathcal{O}\}$ and $\underline{\mathcal{L}} = (\mathcal{C}, \emptyset, \underline{\mathcal{P}})$, where $p_o \notin \mathcal{P}$ and p_o is $(n+1)$-ary if o is n-ary for all $o \in \mathcal{O}$. For each n-ary $o \in \mathcal{O}$, define φ_o to be the formula

$$\forall x_1 \ldots \forall x_{n+2}.\, p_o(x_1, \ldots, x_{n+1}) \rightarrow p_o(x_1, \ldots, x_{n+2}) \rightarrow x_{n+1} = x_{n+2}. \tag{1}$$

φ_o says that p_o is the graph of a partial n-ary function.

Equations 2-7 translate a PFOL formula of $\mathcal{L}$ into a regular formula of $\underline{\mathcal{L}}$. The untranslated formula is underlined, and rewritten until no part of it is underlined.

Equations 2-4 apply to atomic formulas.

$$\underline{p(t_1, \ldots, t_n)} \quad = \quad p(t_1, \ldots, t_n), \tag{2}$$

where each term in $\{t_1, \ldots, t_n\}$ is a variable or a constant.

Operator symbols are eliminated by the following equation.

$$\begin{aligned} &\underline{p(s_1, \ldots, o(t_1, \ldots, t_n), \ldots, s_m)} \\ &\quad = \exists y.\, \underline{p_o(t_1, \ldots, t_n, y)} \wedge \underline{p(s_1, \ldots, y, \ldots, s_m)}, \end{aligned} \tag{3}$$

where $p \in \underline{\mathcal{P}}$, $o \in \mathcal{O}$, and y does not occur in $p(s_1, \ldots, o(t_1, \ldots, t_n), \ldots, s_m)$. Notice that when the application of an operator symbol is undefined, the atomic formula in which it occurs is false. In general, any atomic formula that contains an undefined term is false.

Definite description symbols are eliminated by the following equation.

$$\begin{aligned} &\underline{p(t_1, \ldots, \mathrm{I}x.\, \psi, \ldots, t_n)} \\ &\quad = \exists y.\, \underline{\psi}[x \mapsto y] \wedge (\forall z.\, \underline{\psi}[x \mapsto z] \rightarrow y = z) \wedge \underline{p(t_1, \ldots, y, \ldots, t_n)}, \end{aligned} \tag{4}$$

where $p \in \underline{\mathcal{P}}$, and y does not occur in $p(t_1, \ldots, \mathrm{I}x.\, \psi, \ldots, t_n)$. The definite description term $\mathrm{I}x.\, \psi$ denotes the unique x that satisfies ψ. If there is no single x that satisfies ψ, the term $\mathrm{I}x.\, \psi$ is undefined, and the atomic formula containing the term is false.

The remaining equations apply to non-atomic formulas.

$$\begin{aligned} \underline{\neg\psi} &= \neg\underline{\psi}, & (5) \\ \underline{\psi \rightarrow \varphi} &= \underline{\psi} \rightarrow \underline{\varphi}, & (6) \\ \underline{\forall x.\, \psi} &= \forall x.\, \underline{\psi}. & (7) \end{aligned}$$

Let Γ be a set of formulas over $\mathcal{L} = (\mathcal{C}, \mathcal{O}, \mathcal{P})$. The *regular operator-free* translation of the set is given by

$$\underline{\Gamma} = \{\underline{\varphi} : \varphi \in \Gamma\} \cup \{\varphi_o : o \in \mathcal{O}\}.$$

The structures for full PFOL given in [3] associate a partial function with each operator symbol. Equation 1 ensures that a model for a regular operator-free translation of a set of formulas is a structure for the original set of formulas in the sense defined in [3].

In what follows, the presentation will be simplified by only considering structures that are a model of $\{\varphi_o : o \in \mathcal{O}\}$. As a result, for each predicate symbol p introduced while eliminating an operator symbol, the relation associated with p by every structure $\mathcal{U}$ must be the graph of a partial function.

Definitions for regular formulas over operator-free languages extend to full PFOL as follows. A set of formulas Γ over any language *logically implies* a formula φ iff $\underline{\Gamma} \models \underline{\varphi}$. $\mathcal{U}$ satisfies φ with A iff $\mathcal{U}$ satisfies $\underline{\varphi}$ with A, and $\mathcal{U}$ is a *model* of φ iff $\mathcal{U}$ is a model of $\underline{\varphi}$.

3 Interpretations Between Theories

A *theory* is a set of sentences closed under logical implication. That is, T is a theory iff T is a set of sentences such that for any sentence σ of the language,

$$T \models \sigma \text{ implies } \sigma \in T.$$

Logicians have developed a technique for comparing some theories. When theories T_0 and T_1 are in the same language, T_1 is considered to be as powerful as T_0 if $T_0 \subseteq T_1$. This kind of comparison is possible even when T_0 and T_1 are over different languages, because there may be a way to translate each sentence in T_0 into a sentence in T_1.

Translations that allow the comparison of theories can also be used to combine information from a semantic web. Each source of information supplies a set of sentences Σ. The theory associated with Σ is called the *consequences* of Σ, written $\mathrm{Cn}\,\Sigma$, where

$$\mathrm{Cn}\,\Sigma = \{\sigma : \Sigma \models \sigma\}.$$

The guiding principle is that the consequences of the combined information should be as powerful as the consequences of each of its sources. In other words, the combined information should possess all the inferences available in each of its sources.

Suppose Σ_1 is a set of sentences constructed by combining information from a semantic web, and let Σ_0 be one of the sources of the information. The principle is achieved by finding a translation of each sentence in the consequences of Σ_0 into the language of Σ_1 such that the translated sentence is a member of the consequences of Σ_1.

The presentation of translations between theories is based on [2, §2.7], but modified to handle undefined terms.

The translation of a formula is parameterized by an interpretation. An *interpretation* π of a source language $\mathcal{L}_0 = (\mathcal{C}, \mathcal{O}, \mathcal{P})$ into the destination theory T_1 is a function on the symbols of the language such that

1. π assigns to $\forall$ a formula $\pi_\forall$ of $\mathcal{L}_1$ in which at most one variable occurs free, such that
$$T_1 \models \exists x.\, \pi_\forall[x], \tag{8}$$
where $\varphi[x]$ abbreviates $\varphi[y \mapsto x]$ when only y occurs free in φ.

2. π assigns to each constant symbol $a \in \mathcal{C}$ a closed term π_a of $\mathcal{L}_1$, such that
$$T_1 \models \pi_\forall[\pi_a]. \tag{9}$$

3. π assigns to each n-ary operator symbol $o \in \mathcal{O}$ a term π_o of $\mathcal{L}_1$ in which at most n variables occur free, such that
$$\begin{aligned} T_1 \models \forall x_1 \cdots \forall x_{n+1}.\, \pi_\forall[x_1] \to \cdots \to \pi_\forall[x_n] \\ \to x_{n+1} = \pi_o[x_1, \ldots, x_n] \to \pi_\forall[x_{n+1}]. \end{aligned} \tag{10}$$

4. π assigns to each n-ary predicate symbol $p \in \mathcal{P}$ a formula π_p of $\mathcal{L}_1$ in which at most n variables occur free.

Equations 8–10 are called the *obligations* of the interpretation.

Revisiting the genealogy example, and assuming site A and B agree on the constant terms used to identify people and jobs, an interpretation π of the language of site B into the theory of site A is:

$$\begin{aligned}
\pi_\forall[x] &= (x = x), \\
\pi_{\mathsf{dad}}[y] &= \mathrm{I}x.\, \mathsf{parent_of}(x, y) \wedge \mathsf{is_male}(x), \\
\pi_{\mathsf{mom}}[y] &= \mathrm{I}x.\, \mathsf{parent_of}(x, y) \wedge \mathsf{is_female}(x), \\
\pi_{\mathsf{job}}[x, y] &= \mathsf{employed_as}(x, y).
\end{aligned}$$

Given the interpretation π of the source language $\mathcal{L}_0$ into the destination theory T_1, expressions of $\mathcal{L}_0$ can be translated into expressions of $\mathcal{L}_1$ as follows:

$$\begin{aligned}
\Pi(x) &= x \\
\Pi(a) &= \pi_a \\
\Pi(\mathrm{I}x.\,\varphi) &= \mathrm{I}x.\,\pi_\forall[x] \wedge \Pi(\varphi) \\
\Pi(o(t_1, \ldots, t_n)) &= \pi_o[\Pi(t_1), \ldots, \Pi(t_n)] \\
\Pi(p(t_1, \ldots, t_n)) &= \pi_p[\Pi(t_1), \ldots, \Pi(t_n)] \\
\Pi(\neg\varphi) &= \neg\Pi(\varphi) \\
\Pi(\varphi \to \psi) &= \Pi(\varphi) \to \Pi(\psi) \\
\Pi(\forall x.\,\varphi) &= \forall x.\,\pi_\forall[x] \to \Pi(\varphi)
\end{aligned}$$

The correctness of the translation is addressed next. Let $\mathcal{U}$ be a model of theory T_1. There is a natural way to extract from $\mathcal{U}$ a structure ${}^\pi\mathcal{U}$ for $\mathcal{L}_0$ as follows:

$$\begin{aligned}
{}^\pi\mathcal{U}_\forall &= \text{the set defined in } \mathcal{U} \text{ by } \pi_\forall, \\
{}^\pi\mathcal{U}_a &= \text{the element in } \mathcal{U} \text{ defined by } \pi_a, \\
{}^\pi\mathcal{U}_p &= \text{the relation defined in } \mathcal{U} \text{ by } \pi_p \text{ restricted to } {}^\pi\mathcal{U}_\forall, \\
{}^\pi\mathcal{U}_{p_o} &= \text{the relation defined in } \mathcal{U} \text{ by } \pi_o[x_1, \ldots, x_n] = x_{x+1} \\
& \text{ restricted to } {}^\pi\mathcal{U}_\forall.
\end{aligned}$$

Equation 8 ensures ${}^\pi\mathcal{U}_\forall$ is not empty, and Equation 9 ensures ${}^\pi\mathcal{U}_a$ is a member of ${}^\pi\mathcal{U}_\forall$.

Lemma 1 *Let π be an interpretation of $\mathcal{L}_0$ into T_1, and let $\mathcal{U}$ be a model of T_1. For any formula φ of $\mathcal{L}_0$ and any variable assignment A into ${}^{\pi}\mathcal{U}_{\forall}$, ${}^{\pi}\mathcal{U}$ satisfies φ with A iff $\mathcal{U}$ satisfies $\Pi(\varphi)$ with A.*

Proof. The proof is by structural induction on φ, but only the case of an atomic formula is nontrivial.

Let $\varphi = p(t_1, \ldots, t_n)$, so the translation of φ is

$$\Pi(p(t_1, \ldots, t_n)) = \pi_p[\Pi(t_1), \ldots, \Pi(t_n)].$$

The case in which each term in $\{t_1, \ldots, t_n\}$ is a variable or a constant is easy. The range of the variable assignment A assures $\mathcal{U}$ associates an element of ${}^{\pi}\mathcal{U}$ with each variable. For each constant, Equation 9 makes the same assurance.

Assume φ has the form $p(s_1, \ldots, o(t_1, \ldots, t_n), \ldots, s_m)$. By induction, assume

$${}^{\pi}\mathcal{U} \text{ satisfies } \exists y.\, p_o(t_1, \ldots, t_n, y) \wedge p(s_1, \ldots, y, \ldots, s_m) \text{ with } A$$

iff $\mathcal{U}$ satisfies ψ with A, where

$$\begin{aligned} \psi &= \Pi(\exists y.\, p_o(t_1, \ldots, t_n, y) \wedge p(s_1, \ldots, y, \ldots, s_m)) \\ &= \exists y.\, \pi_{\forall}[y] \wedge y = \pi_o[\Pi(t_1), \ldots, \Pi(t_n)] \wedge \pi_p[\Pi(s_1), \ldots, y, \ldots, \Pi(s_m)], \end{aligned}$$

and $\mathcal{U}$ associates an element of ${}^{\pi}\mathcal{U}$ with each term. Consider ψ', where

$$\begin{aligned} \psi' &= \Pi(p(s_1, \ldots, o(t_1, \ldots, t_n), \ldots, s_m)) \\ &= \pi_p[\Pi(s_1), \ldots, \pi_o[\Pi(t_1), \ldots, \Pi(t_n)], \ldots, \Pi(s_m)] \\ &= \exists y.\, y = \pi_o[\Pi(t_1), \ldots, \Pi(t_n)] \wedge \pi_p[\Pi(s_1), \ldots, y, \ldots, \Pi(s_m)]. \end{aligned}$$

$\mathcal{U}$ satisfies ψ with A iff $\mathcal{U}$ satisfies ψ' with A by Equation 10.

The case of φ having the form $p(t_1, \ldots, \mathrm{I}x.\, \psi, \ldots, t_n)$ is similar to the previous case.

Theorem 1 *For a set of sentences Σ of $\mathcal{L}_0$, $\{\Pi(\sigma) : \sigma \in \Sigma\} \subseteq T_1$ iff for every model $\mathcal{U}$ of T_1, ${}^{\pi}\mathcal{U}$ is a model of Σ.*

Proof. By Lemma 1, for every model $\mathcal{U}$ of T_1, ${}^{\pi}\mathcal{U}$ is a model of Σ iff $\mathcal{U}$ is a model of $\{\Pi(\sigma) : \sigma \in \Sigma\}$ iff $\{\Pi(\sigma) : \sigma \in \Sigma\} \subseteq T_1$.

An *interpretation* π of a theory T_0 into a theory T_1 is an interpretation π of the language of T_0 into T_1 such that

$$\sigma \in T_0 \text{ implies } \Pi(\sigma) \in T_1.$$

An *embedding* π of a set of sentences Σ_0 into a set of sentences Σ_1 is an interpretation π of the theory $\mathrm{Cn}\,\Sigma_0$ into the theory $\mathrm{Cn}\,\Sigma_1$, such that

$$\sigma \in \Sigma_0 \text{ implies } \Pi(\sigma) \in \Sigma_1.$$

Obviously, the obligations of interpretation π are given by

$$\Sigma_1 \models \exists x.\, \pi_\forall[x],$$

for each constant a of the language of Σ_0,

$$\Sigma_1 \models \pi_\forall[\pi_a],$$

and for each operator symbol o of the language of Σ_0,

$$\begin{aligned}\Sigma_1 \models \forall x_1 \cdots \forall x_{n+1}.\, \pi_\forall[x_1] \rightarrow \cdots \rightarrow \pi_\forall[x_n] \\ \rightarrow x_{n+1} = \pi_o[x_1, \ldots, x_n] \rightarrow \pi_\forall[x_{n+1}].\end{aligned}$$

An embedding π of a set of sentences Σ_0 into a set of sentences Σ_1 has the desired properties. By Theorem 1, a model of Σ_0 can be extracted from each model of Σ_1, and therefore, every inference in $\mathrm{Cn}\,\Sigma_0$ is available in $\mathrm{Cn}\,\Sigma_1$.

4 SHOE in PFOL

The Simple HTML Ontology Extensions (SHOE) is an ontology-based knowledge representation language that is embedded in web pages. From [5, Section 2]:

> ... SHOE extends HTML with a set of knowledge oriented tags that, unlike HTML tags, provide structure for knowledge acquisition as opposed to information presentation. SHOE associates meaning with content by making each web page commit to one or more ontologies. These ontologies permit the discovery of implicit knowledge through the use of taxonomies and inference rules, allowing information providers to encode only the necessary information on their web pages, and to use the level of detail that is appropriate to the context. Interoperability is promoted through the sharing and reuse of ontologies.

Consider the task of combining two SHOE semantic data sources that share no ontologies into a single SHOE specification. That task may be evaluated using PFOL as a foundation.

The mapping of a SHOE specification into First-Order Logic given in [5] creates a set of formulas over an operator-free language. Since each formula is regular, every model of the set is a model of the set in PFOL, and vice versa. Furthermore, every SHOE formula is either an atomic formula or a Horn clause. As a result, implementations of SHOE can use inference algorithms developed for Datalog [6].

The data combination task is evaluated by studying the interpretations that are available in the SHOE logic as translated into First-Order Logic.

1. A unary predicate can be associated with the universal quantifier of a source theory. A Horn clause connot serve the same function because the translation would produce formulas not allowed in the SHOE logic.

2. A constant can be translated into another constant. A constant cannot be translated into a term with operators because the SHOE language is operator-free.

3. Since the SHOE language is operator-free, operators cannot be translated.

4. A predicate can be translated into a another predicate. A predicate cannot be translated into a Horn clause if the predicate appears in the hypothesis of another Horn clause.

SHOE provides a limited facility for expressing interpretations. It provides a mechanism for renaming predicate symbols. There is no general mechanism for translating constants, and no mechanism for associating a predicate with the universal quantifier of a source theory. The use of an operator-free language allows efficient inference algorithms, but greatly limits the ways in which information can be combined when it shares no ontologies.

5 Conclusion

Using the method of interpretations between theories in PFOL, this paper established criteria for combining information from a semantic web in a fashion that preserves the inferences available in the original information. The use of PFOL allows translations in which each term is translated into a term, and allows a natural description of specifications that involve undefined terms.

References

[1] Tim Berners-Lee and Mark Fischetti. *Weaving the Web: the original design and ultimate destiny of the World Wide Web by its inventor*. HarperCollins, 1999. Also see http://www.w3.org/DesignIssues/Semantic.html.

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

[3] W. M. Farmer and J. D. Guttman. A set theory with support for partial functions. *Studia Logica*, 65:59–70, 2000. Forthcoming.

[4] J. Heflin, J. Hendler, and S. Luke. SHOE: A knowledge representation language for internet applications. Technical Report CS-TR-4078 (UMIACS TR-99-71), Dept. of Computer Science, University of Maryland, 1999. http://www.cs.umd.edu/projects/plus/SHOE.

[5] Jeff Heflin and James Hendler. Dynamic ontologies on the web. In *Proc. of AAAI*, 2000.

[6] J. D. Ullman. *Principles of Database and Knowledge-Base Systems*, volume I and II. Computer Science Press, Rockville, Maryland, 1989.