Relatedness Predicate Logic

Richard L. Epstein and Stanisław Krajewski[1]

Subject matter relatedness logic, **S**, is designed to model the idea that the subject matter of propositions should be taken into account in reasoning, where the notion of the subject matter of a proposition is taken as primitive. In this paper we extend **S** by considering how the subject matter of a proposition can depend on the subject matter of its parts when propositions are parsed in predicate logic. This is an example of a general method for how to incorporate a semantic aspect of propositions other than truth and reference into a predicate logic.

The propositional logic S

The propositional logic **S** was first presented in Epstein, 1979. It was further developed with a strongly complete axiomatization and clearer motivation in Epstein, 1990, from which this brief description is culled.

A *subject matter relatedness model* is:

$$L(\neg, \rightarrow, p_0, p_1, \ldots)$$

$\downarrow$ realization

$\{p_0, p_1, \ldots,$ complex propositions formed from these using $\neg, \rightarrow\}$

$\downarrow$ v, r, and truth-tables

$$\{T, F\}$$

Here $p_0, p_1, \ldots$ are the realizations of $p_0, p_1, \ldots$, which are taken to be atomic. They are English language sentences; from a realist perspective these would be understood as corresponding to or representing abstract propositions. Then v is an assignment of truth-values to the atomic propositions. And R is a symmetric, reflexive relation on pairs of atomic propositions meant to be understood as p has *subject matter overlap* with q. The extension of R to all complex propositions is given inductively by the **S**-*conditions*:

R1. R(A,A)
R2. R(A,B) iff R(¬A,B)

[1] The work here was originally developed by the authors when Epstein was an Exchange Scholar of the U. S. National Academy of Sciences to the Polish Academy of Sciences in Warsaw in 1981. It was the subject of a series of lectures by Epstein to the Logic Group at Iowa State University in 1982, when the first draft of this paper was written. We are grateful to Roger Maddux, Donald Pigozzi, Howard Blair, and William Robinson who offered much help in developing these ideas at that time. This research was earlier reported on in Krajewski, 1986. The present paper is a revised version of a talk given at the 3rd Conference on Logic and Reasoning of the Advanced Reasoning Forum, Berkeley, 2001.

© 2004 Richard L. Epstein and Stanisław Krajewski.
Bulletin of Advanced Reasoning and Knowledge 2 (2004) 19–38.

R3. $\mathsf{R}(A,B)$ iff $\mathsf{R}(B,A)$
R4. $\mathsf{R}(A,B\to C)$ iff $\mathsf{R}(A,B)$ or $\mathsf{R}(A,C)$

Lemma 1 *a*. The only transitive relatedness relation is the universal relation.
b. For any relatedness relation R,
$\mathsf{R}(A, B)$ iff for some p_i in A, some p_j in B, $\mathsf{R}(p_i, p_j)$.

Proof For part (a), note that for any A, B, both $\mathsf{R}(A, A\to B)$ and $\mathsf{R}(A\to B, B)$. So if R is transitive, $\mathsf{R}(A, B)$. Part (b) can be proved by double induction on the length of the formulas A and B.[2] ■

Thus, any symmetric, reflexive relation r on atomic propositions determines a relatedness relation R on all formulas via the condition of Lemma 1.b.

Then v is extended inductively to all propositions by using the classical table for $\neg$ and the table for the related conditional for $\to$, namely:

A	B	$\mathsf{R}(A,B)$	$A\to B$
any	value	fails	F
T	T	holds	T
T	F	holds	F
F	T	holds	T
F	F	holds	T

A proposition A of the semi-formal language is *true* in the model if $\mathsf{v}(A) = \mathsf{T}$, *false* if $\mathsf{v}(A) = \mathsf{F}$. We refer to a model as $\mathsf{M} = \langle \mathsf{v},\mathsf{R}\rangle$ and write $\mathsf{M}\vDash A$ for $\mathsf{v}(A) = \mathsf{T}$. Letting capital Greek letters stand for collections of propositions (atomic or compound), we define the *semantic consequence relation of* **S**.

$\Gamma \vDash_{\mathbf{S}} A$ iff for every model M, if for every B in Γ, $\mathsf{M}\vDash B$, then $\mathsf{M}\vDash A$

Since the universal relation can be a relatedness relation, the *logic* **S** *is a sublogic of the classical propositional logic*: If $\Gamma \vDash_{\mathbf{S}} A$ then $\Gamma \vDash_{\text{classical}} A$.

In **S** we define the following connectives:

$R(A, B) \equiv_{\text{Def}} A\to(B\to B)$

$A\wedge B \equiv_{\text{Def}} \neg(A\to(B\to \neg((A\to B)\to(A\to B))))$

Then:

$\langle \mathsf{v},\mathsf{R}\rangle \vDash R(A,B)$ iff $\mathsf{R}(A, B)$

$\langle \mathsf{v},\mathsf{R}\rangle \vDash A\wedge B$ iff $\langle \mathsf{v},\mathsf{R}\rangle \vDash A$ and $\langle \mathsf{v},\mathsf{R}\rangle \vDash B$

The connective $\vee$ can be defined from $\neg$ and $\to$ as either classical disjunction or as relatedness disjunction. Since that decision is inessential and distracting, we will not consider disjunction in the discussions below.

Using these definitions a strongly complete axiomatization of **S** is given in Epstein, 1990. That is, letting $\vdash_{\mathbf{S}}$ stand for the syntactic consequence relation:

For every Γ, A, $\Gamma \vDash_{\mathbf{S}} A$ iff $\Gamma \vdash_{\mathbf{S}} A$.

---

2 See p. 99 of Epstein, 1990.

Instead of starting with a subject matter relatedness relation, we can take the notion of a subject matter as primitive and then define two propositions being related if they have some subject matter in common. We first postulate a set of topics that we assign to the propositions under consideration and have:

A set of topics $\mathsf{S} \neq \varnothing$.

An assignment s that for every atomic proposition p, $\mathsf{s}(p) \subseteq \mathsf{S}$ and $\mathsf{s}(p) \neq \varnothing$.

The extension of s to complex propositions is given by taking the *subject matter of a proposition to be the sum of the subject matter of its parts*:

$$\mathsf{s}(A) = \bigcup \{\mathsf{s}(p) : p \text{ appears in } A\}$$

Any such s and S we call a *subject matter assignment*.

The *relatedness relation associated with* s is:

$$\mathsf{R}_\mathsf{s}(A, B) \text{ iff } \mathsf{s}(A) \cap \mathsf{s}(B) \neq \varnothing$$

Given any subject matter relatedness relation, R, we can define the subject matter of a proposition, A, to be $\mathsf{s}_\mathsf{R}(A) = \{\{A, B\}: \mathsf{R}(A, B)\}$. This is the *subject matter assignment associated with* R.

Lemma 2 *a*. For any relatedness relation R, $\mathsf{R}(A, B)$ iff $\mathsf{s}_\mathsf{R}(A) \cap \mathsf{s}_\mathsf{R}(B) \neq \varnothing$.

b. For any subject matter assignment s, R_s satisfies R1–R4.

c. For any relatedness relation R, $\mathsf{R} = \mathsf{R}_{\mathsf{s}_\mathsf{R}}$.

d. For any subject matter assignment s,
$\mathsf{s}(A) \cap \mathsf{s}(B) \neq \varnothing$ iff $\mathsf{s}_{\mathsf{R}_\mathsf{s}}(A) \cap \mathsf{s}_{\mathsf{R}_\mathsf{s}}(B) \neq \varnothing$.

Proof a. $\mathsf{s}_\mathsf{R}(A) \cap \mathsf{s}_\mathsf{R}(B) \neq \varnothing$ iff $\{\{A, C\}: \mathsf{R}(A, C)\} \cap \{\{B, D\}: \mathsf{R}(B, D)\} \neq \varnothing$

iff some C, D, $\{A, C\} = \{B, D\}$

iff either $A = B$ (and so $\mathsf{R}(A, B)$ since R is reflexive)

or some C, $C = B$ (and so $\mathsf{R}(A, B)$)

or some D, $D = A$ (and so $\mathsf{R}(A, B)$, since R is symmetric).

b. Because for every p, $\mathsf{s}(p) \neq \varnothing$, we have $\mathsf{s}(A) \neq \varnothing$ for every A, so $\mathsf{R}(A, A)$ holds. And R is symmetric. We also have $\mathsf{s}(A) = \mathsf{s}(\neg A)$, so $\mathsf{R}(A, B)$ iff $\mathsf{R}(\neg A, B)$. And $\mathsf{s}(A \to B) = \mathsf{s}(A) \cup \mathsf{s}(B)$ so $\mathsf{R}(A, B \to C)$ iff $\mathsf{R}(A, B \wedge C)$. Finally,

$$\begin{aligned} \mathsf{R}(A, B \to C) \text{ holds } &\text{iff } \mathsf{s}(A) \cap [\mathsf{s}(B) \cup \mathsf{s}(C)] \neq \varnothing \\ &\text{iff } [\mathsf{s}(A) \cap \mathsf{s}(B)] \cup [\mathsf{s}(A) \cap \mathsf{s}(C)] \neq \varnothing \\ &\text{iff } \mathsf{s}(A) \cap \mathsf{s}(B) \neq \varnothing \text{ or } \mathsf{s}(A) \cap \mathsf{s}(C) \neq \varnothing \\ &\text{iff } \mathsf{R}(A, B) \text{ or } \mathsf{R}(A, C) \end{aligned}$$

c. $\mathsf{R}_{\mathsf{s}_\mathsf{R}}(A, B)$ iff $\mathsf{s}_\mathsf{R}(A) \cap \mathsf{s}_\mathsf{R}(B) \neq \varnothing$ iff there are C, D such that $\{A, C\} \in \mathsf{s}(A)$, $\{B, D\} \in \mathsf{s}(B)$, $\{A, C\} = \{B, D\}$, $\mathsf{R}(A, C)$ and $\mathsf{R}(B, D)$ iff either $A = B$, so $\mathsf{R}(A, B)$, or $A \neq B$, so $C = B$ and $\mathsf{R}(A, B)$.

d. This follows from (a) via the relatedness relation associated with s_R. ■

Since the only aspects of an atomic proposition that are significant in our model are its truth-value and subject matter (in terms of its place in a relatedness relation) we can make an *abstraction of a model*:

$L(\neg, \rightarrow, p_0, p_1, \ldots)$

$\downarrow$ $\quad$ v, R, and truth-tables

$\{T, F\}$

Here v is a way to assign truth-values to the propositional variables; R is a symmetric and reflexive relation on the variables that is extended to all wffs by R1–R4. And v is extended to all wffs by the tables for classical $\neg$ and the related conditional for $\rightarrow$. Any difference between two models is ignored if they result in the same abstract models.

Up to this point we have made no infinitistic assumptions about either the formal language or the semantics. To have full generality and to freely use mathematics in our studies, the following assumption is often invoked.

The fully general relatedness abstraction Any function from the collection of propositional variables to $\{T, F\}$, together with any symmetric, reflexive binary relation on the propositional variables form a model, when the relation is extended to all wffs by R1–R4 and the truth-assignment is extended to all wffs by the truth-tables. Any such relation is called a (*subject matter*) *relatedness relation*.

That is, not only do differences between models not matter if the models result in the same abstract model, we no longer are concerned if an abstract model arises from an actual model.

Classical predicate logic

The notions of realization and model in classical predicate logic depend on how we understand the relation of the formal language of predicate logic to ordinary language. A full presentation of those ideas is given in Epstein, 1994. Here we briefly summarize what is needed from that volume for the development of predicate relatedness logic.

Our formal language is:

$L(\neg, \rightarrow, \forall, P_0, P_1, \ldots, c_0, c_1, \ldots)$

Here each of the logical symbols are taken as primitive: $P_0, P_1, \ldots$ are predicate symbols (where superscripts for the -arity of each have been supressed), $c_0, c_1, \ldots$ are name symbols. Though not explicit in this notation, the formal language also comes equipped with variables, $x_0, x_1, \ldots$. The *meta-variables* t, u, v stand for any term (variables or name symbols); A, B, C, . . . stand for any formulas; i, j, k, m, n stand for natural numbers. We also use x, y, z, w and $y_0, y_1, \ldots$ as variables in informal formulas and as metavariables for variables, and $\vec{x}, \vec{y}, \vec{z}$ to range over sequences of variables. We set:

$\exists x \, A \equiv_{Def} \neg\forall\neg x \, A$

Given a formula A with $x_{i_1}, \ldots, x_{i_n}$ a list of all variables that occur free in A, with $i_1 < \ldots < i_n$, we define the *closure* or *universal closure* of A as:

$\forall \ldots A \equiv_{Def} \forall x_{i_1} \ldots \forall x_{i_n} A$

A *realization* is a *semi-formal language,* which we call *semi-formal English* or *formalized English*. Predicate symbols are realized as linguistic predicates, name symbols by names, and complex formulas are realized by replacing the predicate and names symbols in them by their realizations. For example:

(1) $L(\neg, \rightarrow, \forall, P_0, P_1, \ldots, c_0, c_1, \ldots)$

$\downarrow$

$L(\neg, \rightarrow, \forall, \exists$; 'is a dog', 'is a cat', 'eats grass', 'is a wombat', 'is the father of'; 'Ralph', 'Dusty', 'Howie', 'Juney')

Here 'is a dog' is the realization of P_0^1, which we write as real(P_0^1) = 'is a dog', and 'is a wombat' = real(P_3^1), and 'is the father of' = real(P_0^2). Similarly, 'Howie' realizes c_2, which we write as 'Howie' = real(c_2). The expressions or formulas of the semi-formal language are the realizations of the formal wffs. For example, 'Ralph is a dog' is an expression of the semi-formal language. It is the realization of: $P_0^1(c_0)$. We notate this as real($P_0^1(c_0)$) = 'Ralph is a dog'. Similarly, we have the following expressions of the semi-formal language:

$\neg$(Ralph is a dog) $\rightarrow$ $\forall x_0$ (x_0 is a cat) x_{32} is a dog

These are realizations of, respectively, $\neg P_0^1(c_0) \rightarrow \forall x_0 (P_1^1(x_0))$ and $P_0^1(x_{32})$.

Since predicates and names in a presentation of a realization are pieces of language, we simplify the notation of realization (1) to:

$L(\neg, \rightarrow, \wedge, \forall, P_0, P_1, \ldots, c_0, c_1, \ldots)$

$\downarrow$

$L(\neg, \rightarrow, \wedge, \forall$; is a dog, is a cat, eats grass, is a wombat, is the father of; Ralph, Dusty, Howie, Juney)

In what follows we will use P, Q to range over $\{\text{real}(P_i) : i \geq 0\}$.

A semi-formal language is linguistic, a formalized fragment of English. The predicates and names have their usual meanings in English. So, the truth-values of the atomic propositions are fixed in the realization, though we might not know those truth-values

To determine the truth-values of quantified sentences, however, we have to agree on exactly what objects the variables can range over. We collect all the referents of the names plus any other objects we want the variables to range over and call that the *universe*. For the previous example of a realization we could take the universe to be all animals.

A key assumption of classical predicate logic is that the only semantic property of any name is its reference: How we name objects does not matter.

The extensionality of predicates Given any predicate P and any terms t and u that refer to the same object (possibly through some temporary indication of reference), then P(t) and P(u) have the same truth-value; and similarly for n-ary predicates.

The application of a predicate to an object A *predicate* P *applies to* or *is true of* a particular object iff given a variable x with an indication that x is to refer to the object, then P(x) is true; and similarly for n-ary predicates.

Thus, once we have a realization with a universe, all atomic predications are fixed, though we might not know the truth-values of those.

A model is the extension of the valuation of atomic predications to all closed formulas of the semi-formal language via:

$v(\neg A) = \mathsf{T}$ iff $v \nvDash A$
$v(A \rightarrow B) = \mathsf{T}$ iff not ($v \vDash A$ and $v \nvDash B$)
$v(\forall x\, A) = \mathsf{T}$ iff no matter what object x is taken to refer to, $v(A(x)) = \mathsf{T}$

These clauses can be formulated in terms of assignments of references. Let $\sigma, \tau, \ldots$ range over ways that objects of the universe can be named, either with names from the semi-formal language or temporarily with variables. In most general terms, σ assigns for each variable x an object $\sigma(x)$ of the universe, and for each name c, for every σ and τ, $\sigma(c) = \tau(c)$. We assume that there are sufficient ways to name objects.[3] We make the following definition for each σ.

Recursive definition of truth in a model

$v_\sigma(P(t_0, \ldots, t_n)) = \mathsf{T}$ iff real (P) applied to $\sigma(t_0), \ldots, \sigma(t_n)$ via the (possibly temporary) names $t_0, \ldots, t_n$ is true
$v_\sigma(\neg A) = \mathsf{T}$ iff $v \nvDash A$
$v_\sigma(A \rightarrow B) = \mathsf{T}$ iff not ($v_\sigma \vDash A$ and $v_\sigma \nvDash B$)
(2) $v_\sigma(\forall x\, A) = \mathsf{T}$ iff for every assignment of references τ that differs from σ at most in what it assigns as reference to x, $v_\tau(A) = \mathsf{T}$

For every closed wff A, $v(A) = \mathsf{T}$ iff for every σ, $v_\sigma(A) = \mathsf{T}$.

We also allow for different models to be built on the same realization and universe because we might not know the truth-values of the atomic predications.

In doing the metamathematics of logic we often make the *extensional abstraction* of a classical predicate logic model:

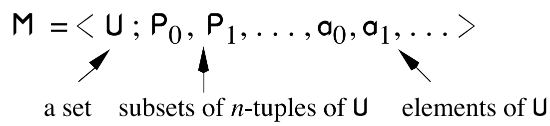
The universe is taken to be a non-empty set.
Each n-ary predicate is identified with a subset of n-tuples of the universe, which we call the *extension* of that predicate.
Each name is taken to refer to one element of the universe.

For example, if the universe of the model is all animals, the predicate 'is a dog' could be identified with the set of dogs, the predicate 'is the father of' could be identified with all pairs of elements of the universe such that the first element is the father of the second element, and 'Juney' could be mapped to the particular dog whose name is 'Juney'.

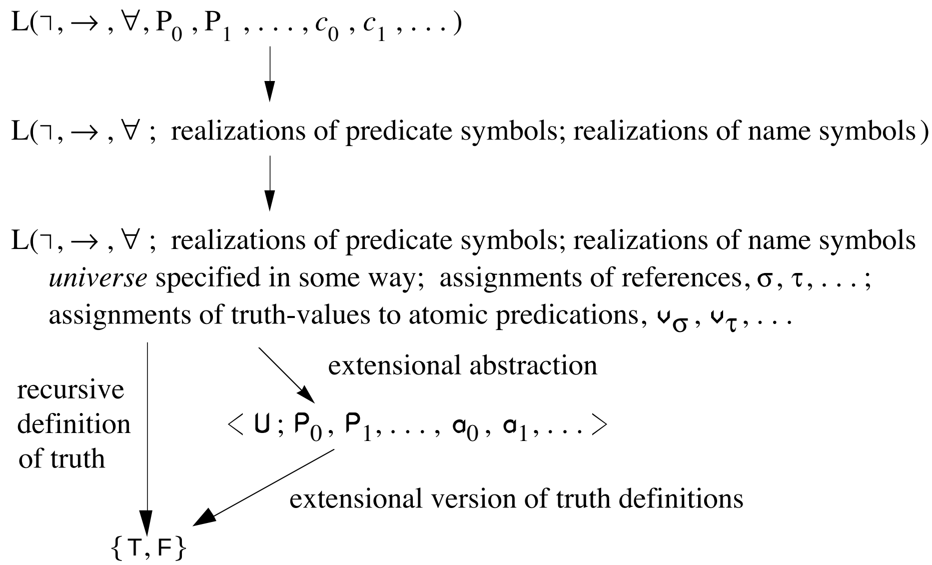
That is, a model is abstracted to:

---

[3] See Epstein, 1994, p. 95.



In an extensional abstraction of a model, a predicate applying to a sequence of objects is true iff the sequence of objects in the universe is in the set of *n*-tuples corresponding to that predicate. We have the following picture:



In doing the metamathematics of logic we often make the *fully general abstraction of classical predicate logic*: Any set $\mathsf{U} \neq \varnothing$, and any subsets of *n*-tuples $\mathsf{P}_0, \mathsf{P}_1, \ldots$ of U, and elements $\mathsf{a}_0, \mathsf{a}_1, \ldots$ of U can make up an extensional model. That is, as with abstract models for propositional logic, any difference between two models is ignored if they result in the same extensional model, and we do not care if an extensional model arises from a realization.

Though a realist could say that linguistic predicates in our realizations are just standing for actual abstract predicates, it is not clear what the name symbols could be realized as if not names. By bypassing entirely linguistic realizations and considering only extensional models, a realist interpretation is possible, though even there, for the example above, the model would be an odd mixture of abstract predicates and concrete objects, such as Juney and Ralph.

The extensional abstraction assumes that all that is significant about a predicate is (if it's a unary one) the set of elements of the universe to which it applies. Hence, consider two predicates:

'is a dog'
'is a cat'
universe: all objects from Earth on Mars

Since these two predicates agree on their extensions, classical predicate logic does not distinguish between them in the sense that they give rise to the same extensional interpretations in the model. Nonetheless, no one believes that 'is a dog' and 'is a cat' are the same predicates in the realization. It is simply an identification of them that we effect in going from the realization to the extensional model, "throwing away what we don't need" (or what we don't know how to use). We treat them in the extensional abstraction as if they were the same predicate. Thus, even though two models may differ in their realizations, if they agree in their extensional abstractions, then they are the same for the mathematical theory. The extensional abstraction supposes that all that is significant about realizations has been captured in the extensional models.

Pure predicate relatedness logic, PS

In building a predicate logic on the propositional logic **S**, we will modify classical models to take into account subject matter relatedness. We begin with the language without names, $L(\neg, \rightarrow, \forall, P_0, P_1, \ldots)$.

In order to give a definition of truth for a realization, we need to know not only the atomic predications, but also what each formula is related to in subject matter. In the propositional case we treated the logical symbols as neutral for relatedness: $\neg$ and $\rightarrow$ contribute nothing to the relation. So in building our simplest model of the notion of relatedness, we will treat the logical symbols $\forall x_i$ and x_i as contributing nothing, too.

In propositional logic the subject matter of a proposition is the union of the subject matters of its parts. We do the same for this predicate logic. We assume for every model the following:

A set of topics $\mathsf{S} \neq \varnothing$.

An assignment s that for every predicate P, $\mathsf{s}(p) \subseteq \mathsf{S}$ and $\mathsf{s}(p) \neq \varnothing$.

$\mathsf{s}(A) = \bigcup \{\mathsf{s}(P) : P \text{ appears in } A\}$

Then relatedness is determined by:

$\mathsf{R}_{\mathsf{s}}(A, B)$ iff $\mathsf{s}(A) \cap \mathsf{s}(B) \neq \varnothing$.

Again, R_{s} is reflexive and symmetric. The definition of truth in a model is as in the classical case except for the clause governing conditionals:

$\mathsf{v}_{\sigma}(A \rightarrow B) = \mathsf{T}$ iff not ($\mathsf{v} \vDash A$ and $\mathsf{v} \nvDash B$) and $\mathsf{R}(A, B)$

The logic **PS** is the semantic consequence relation of all such models in $L(\neg, \rightarrow, \forall, P_0, P_1, \ldots)$. Since the universal relation on wffs is a relatedness relation, *the logic* **PS** *is a sublogic of classical logic*.

In our models, for each predicate we have to specify both a subject matter and the truth-value of each of its atomic predications. For example, we may have two predicates 'is a woodchuck' and 'is a groundhog' which are true of exactly the same objects in all universes. But we may choose to have them related to different predicates, say the first to 'is a silly poem' and the second to 'is a day of the year.'[4] These predicates, then, have the same extension in every model but are not synonymous.

[4] For non-Yankees, 'How much wood would a woodchuck chuck if a woodchuck could chuck wood? As much wood as a woodchuck would chuck if a woodchuck could chuck wood'; and February 2 is 'Groundhog Day.'

At what point is the assignment of subject matters made: in the realization, or in the model (realization plus universe)? It would seem that the subject matter of a predicate is independent of the choice of universe, since subject matters are aspects of the linguistic predicates, that is, of parts of the semi-formal language.

On the other hand, one might say that the subject matter of a predicate depends on what kind of things we are talking about. For example, the subject matter of ‘has two legs’ might be different if the universe is all pieces of furniture as opposed to all living mammals. But this seems to confuse subject matter with the extension of a predicate. What kind of things we are talking about is not something that we can represent in the semi-formal language, aside from using the predicates of the language. For example, if the universe is all mammals, we can express in the semi-formal language that we are talking about all two-legged things by using the predicate ‘has two legs’. But we cannot express that we are talking about all things that have two legs and are mammals. That is implicit, but not expressible in the semi-formal language.

Hence, as the atomic predications that use names are fixed at the realization, so, too, are the subject matters of atomic predicates. But just as we may not know those truth-values, and hence allow different models to be built on what is the same linguistic realization, we allow different models to take account of different choices of subject matter for the same linguistic realization.

As in the propositional case, we could take subject matter relatedness relations as primitive. Given a realization, a *pure predicate subject matter relatedness relation* (or **PS**-*relatedness relation*) is any reflexive, symmetric relation R on formulas satisfying the following.

The **S**-conditions R1–R4.

R5 $\mathsf{R}(P(\vec{x}), P(\vec{y}))$

R6 $\mathsf{R}(P(\vec{x}), Q(\vec{y}))$ iff $\mathsf{R}(P(\vec{z}), Q(\vec{w}))$

R7 $\mathsf{R}(\forall x\, A, B)$ iff $\mathsf{R}(A, B)$

Lemma 3 For any relatedness relation R, $\mathsf{R}(A, B)$ iff for some P in A, some Q in B, some $\vec{x}$ and $\vec{y}$, $\mathsf{R}(P(\vec{x}), Q(\vec{y}))$.

Proof The proof is similar to that for Lemma 1.b. Condition R6 ensures the lemma if A and B both have length 1. Suppose the lemma is true for all A of length $\leq n$ and all Q. To show it holds for all A of length $\leq n+1$ and all Q, all the induction steps are as for Lemma 1, except now $\mathsf{R}(\forall x\, A, Q(\vec{y}))$ iff $\mathsf{R}(A, Q(\vec{y}))$ by R7, and hence by induction, $\mathsf{R}(\forall x\, A, Q(\vec{y}))$ iff $\mathsf{R}(P(\vec{x}), Q(\vec{y}))$ for some P in A. The rest of the proof we leave to you. ■

Thus, any symmetric, reflexive relation r on atomic predicates determines a pure relatedness relation R on all formulas via the condition of Lemma 3 and $\mathsf{R}(P(\vec{x}), Q(\vec{y}))$ iff $\mathsf{r}(P, Q)$.

Given R we may define $\mathsf{s}_{\mathsf{R}}(P) = \{\{P, Q\}: \mathsf{R}(P(\vec{x}), Q(\vec{y}))\}$. The proof of the following is a modification of the proof of Lemma 2 that we leave to you.

Lemma 4 *a*. For any subject matter assignment s, $\mathsf{R_s}$ satisfies R1–R7.
b. $\mathsf{s}(A) \cap \mathsf{s}(A) \neq \varnothing$ iff $\mathsf{R_s}(A, B)$ iff $\mathsf{s}_{\mathsf{R_s}}(A) \cap \mathsf{s}_{\mathsf{R_s}}(A) \neq \varnothing$.
c. $\mathsf{R}(A, B)$ iff $\mathsf{s_R}(A) \cap \mathsf{s_R}(A) \neq \varnothing$ iff $\mathsf{R}_{\mathsf{s_R}}(A, B)$.

We can characterize syntactically the relatedness relation of a model:

$$R(A, B) \equiv_{Def} \forall \ldots (A \rightarrow (B \rightarrow B))$$

If A and B are closed wffs, then R(A, B) is $A \rightarrow (B \rightarrow B)$, as in the propositional case. We can also define as for the propositional logic:

$$A \wedge B \equiv_{Def} \neg(A \rightarrow (B \rightarrow \neg((A \rightarrow B) \rightarrow (A \rightarrow B))))$$

We leave to you to show that in any model:

$\mathsf{v}_\sigma \vDash R(A, B)$ iff $\mathsf{R}\,(A, B)$ in M

$\mathsf{v}_\sigma \vDash A \wedge B$ iff $\mathsf{v}_\sigma \vDash A$ and $\mathsf{v}_\sigma \vDash B$

An *axiomatization of pure relatedness predicate logic* is given by:

Take an axiomatization of classical predicate logic in which the only rule is *modus ponens* for closed formulas.[5]

Replace the propositional axiom schema by the axiom schema of **S**.

Add the following axiom schema:

1. $R(\, P(\vec{x}), P(\vec{y})\,)$
2. $R(\, P(\vec{x}), Q(\vec{y})) \rightarrow R(\, P(\vec{z}\,), Q(\vec{w})\,)$
3. $R(\forall x\, A, B) \leftrightarrow R(A, B)$

A straightforward modification of the usual Henkin-style completeness proof for predicate logic and the completeness proof for **S** will show that this axiomatization is strongly complete. That is, writing '$\vdash$' for syntactic consequence, $\Gamma \vdash A$ iff $\Gamma \vDash A$. The proof of this and other completeness theorems we discuss below will appear in a sequel to this paper.

An example

Consider the proposition:

Numbers cough.

In predicate logic we would formalize this as:[6]

(3) $\forall x\, (x$ is a number $\rightarrow$ x coughs$)$

This is true in any classical predicate logic model whose universe is composed of all animals, since the antecedent is false of every object.

But is there subject matter overlap in (3)? At least on the ordinary reading of the words, we normally would say that there is no subject matter in common between 'number' and 'coughs': It's a category mistake to predicate both

[5] As in, for example, Grzegorczyk, 1974, or more particularly as in Epstein, 2005.
[6] See Chapter V of Epstein, 1994.

'coughs' and 'is a number' of anything. We don't even need to know what the universe of the model is to know the following.

(4) $\mathsf{s}\,(\text{is a number}) \cap \mathsf{s}\,(\text{coughs}) = \varnothing$

Hence (3) is false.

How we determine the categories is not crucial to this example. It's enough that people in general think that (4) is right, and philosophers in particular talk about category mistakes. We apparently have semantic categories implicit in our talk. We needn't assume that categories and subject matter are part of our ontology. We can view them as convenient ways of speaking about usage.

Now either we accept that such talk makes sense, and try to model it; or we show that such talk doesn't make sense, perhaps by showing the consequences of attempting to model it.

Still, someone might say that 'numbers' and 'coughs' do have something in common. We use numbers to count, and we can count coughs. That is 'numbers' and 'counts' are related, and 'counts' and 'coughs' are related. So 'numbers' and 'coughs' are related. But remember that we are talking here of immediate subject matter overlap, one-step relatedness. The notion of non-empty intersection is not transitive. Just as in Lemma 1, if subject matter relatedness is taken to be transitive, predicate relatedness logic collapses into classical logic because every proposition will be related to every other. We can always cook up connections.

Nor is it right to say that 'numbers' and 'coughs' are related because there is a chain of reasoning that can get us from talking about numbers to talking about coughing. That would be to invoke an entirely different notion of relatedness, something like deductive relevance.

In any case, we do not see there being one subject matter model of ordinary language. Rather, we agree on the categories of natural language we are using, and that determines the model.

This is not intended to be a full analysis. It raises difficult questions: What does 'make sense' mean? How do we set up categories? Given the categories, how do we decide which predicates go into which categories? Are all our models psychological in a way that, say, assigning truth to an 'if . . . then . . .' claim in classical logic isn't?

What we have done here is develop tools to investigate these questions, based on the assumption that we can assign subject matters to predicates and names.

Equality without names

In **PS** we could designate, say, P_0 to play the role of equality. We could add the usual axioms for '=', except now for P_0, and specify the properties we wish equality to have for relatedness by adding additional axioms governing P_0, which by the strong completeness theorems is equivalent to restricting the class of models appropriately. But one property we cannot specify is that P_0 is the identity of the universe and not an equivalence relation. That we can do only if we agree to it by convention, as is usual when adding ‘=’ to the language of predicate logic.[7]

So for relatedness predicate logic, too, we add ‘=’ to the language, stipulating that in all models it is to be realized as the identity on the universe. That is, for any assignment σ, $v_\sigma \vDash x = y$ iff $\sigma(x)$ is the same object in the universe as $\sigma(y)$.

We need to specify how we will treat ‘=’ for relatedness. We consider four choices, for each of which we will continue to have that the subject matter of a proposition is the union of the subject matters of its parts:

$$\mathsf{s}(\mathrm{A}) = \bigcup\{\mathsf{s}(\mathrm{P}) : \mathrm{P} \text{ in A}\} \cup \bigcup\{\mathsf{s}(c) : c \text{ in A}\} \cup \{\mathsf{s}(\text{‘='}) : \text{‘=' is in A}\}$$

a. We could treat ‘=’ as any other predicate: It is related to itself and, depending on the model, related to some other predicates, unrelated to others. In terms of subject matter assignments, the only requirement we have is that $\varnothing \neq \mathsf{s}(\text{‘='}) \subseteq \mathsf{S}$.

But in distinguishing ‘=’ as the identity of the universe we are elevating it to the status of a logical symbol, as discussed in Epstein, 1994. The interpretation of it is always the same, regardless of the model. And so its place in the relatedness relation should always be the same, regardless of the model. We reject this choice.

b. We could say that just as $\neg$, $\rightarrow$, and $\forall x$ contribute nothing to the subject matter of a formula in which they occur, ‘=’ has no subject matter, too. Equality would be completely neutral with respect to subject matter, adding nothing to the topics. Thus, if ‘x is a prime number’ is unrelated to ‘y is a dog’, then ‘x is a prime number and $x = x$’ is unrelated to ‘y is a dog and $x = x$.’ In terms of subject matter assignments this amounts to taking $\mathsf{s}(\text{‘='}) = \varnothing$.

But then $\forall x\, (x = x \rightarrow x = x)$ would be false in every model. However, for every unary predicate P, the following is valid:

$$\forall x \forall y\, ((x = x \wedge \mathrm{P}(x) \rightarrow \mathrm{P}(x)) \rightarrow (y = y \wedge \mathrm{P}(y) \rightarrow \mathrm{P}(y)))$$

We would not be able to reason directly about equality. We could reason about equality only after we have established a particular topic. Or we could reason directly about equality using $\wedge$ and classical $\neg$, except that we would have to take $\wedge$ as primitive. The definition of classical $\wedge$ as in Section A no longer would work, for it would give as valid $\forall x\, \forall y\, (x = y \wedge \mathrm{P}(x))$. This seems too large a departure from what we have done previously to develop here.

c. We could say that ‘=’, since it is a predicate, should be related to itself in order to preserve the reflexivity of the relatedness relation; but because it is a logical symbol, it should not be related to any other predicate. In terms of subject matters, we add a distinguished element e to every set of topics, $e \in \mathsf{S}$, and set $\mathsf{s}(\text{‘='}) = \{e\}$, while for every other predicate P, $e \notin \mathsf{s}(\mathrm{P})$. In terms of the relatedness relation, we would add the conditions:

(5) $\mathsf{R}(x = y,\ z = w)$

not $\mathsf{R}(x = y,\ \mathrm{P}(\vec{z}))$

---

[7] See Chapter VI of Epstein, 1994.

The latter is the first negative condition imposed on relatedness used in establishing a logic.

We then take *the logic* **PS(=)** to be the semantic consequence relation of all **PS** models in which (5) holds and '=' is evaluated as the identity.

Now the usual axioms for equality hold, for example, $x = y \rightarrow y = x$. But the following will not be valid.

(6) $\forall x \forall y\, (x = y \rightarrow (\mathrm{A}(x) \leftrightarrow \mathrm{A}(y)))$

Rather, defining $\wedge$ as before, the following is valid.

(7) $\forall x \forall y\, ((x = y \wedge \mathrm{A}(x)) \rightarrow \mathrm{A}(y))$

Equality cannot stand apart from other predicates when we wish to invoke it in relation to those predicates, but must be in a clause appended to the predicate.

By adding the syntactic equivalents of (5) to **PS**, along with the usual axioms governing '=' from the classical predicate logic modified to use (7) in place of (6), we can obtain a strongly complete axiomatization of **PS(=)**. But what we don't have is that the universal relation on formulas is a relatedness relation. Hence, we don't have that **PS(=)** is a sublogic of classical logic. In particular, the negation of (6) is valid in this system for every A in which '=' does not occur.

d. We can say that '=', like other logical symbols, should be neutral in the sense that it can be used in any formula without involving issues of relatedness. This we can accomplish by taking '=' to be related to every predicate. In terms of subject matter assignments, s('=') = S. In terms of the relatedness relation, we have the following requirement:

(8) $\mathsf{R}(x = y, \mathrm{A})$

The logic **PS(=$_\mathbf{u}$)** is the semantic consequence relation of all **PS** models in which (8) holds and '=' is evaluated as the identity. Since the universal relation on formulas is now a relatedness relation, **PS(=$_\mathbf{u}$)** is a sublogic of classical predicate logic with equality.

To axiomatize **PS(=$_\mathbf{u}$)**, we need add only the scheme R($x = y$, A) and the classical axiom schema for equality to the axiom system for **PS**.

But now we have that the following is valid:

(9) $\forall x\, ((x = x \wedge x \neq x) \rightarrow \forall x\, \mathrm{P}(x))$

For example, $\forall x\, ((x = x \wedge x \neq x) \rightarrow \forall x\, (x \text{ is a dog}))$ is true. Doesn't this violate the intuition on which **S** was based? Why should this be true, yet 'The moon is made of green cheese $\rightarrow$ 2 + 2 = 4' is false?

That (9) is valid just reflects our view of the universal relevance of equality to all our reasoning. If you find this unacceptable, then option **PS(=)** will be preferable. The choice between (9) being valid or the negation of (6) being valid reflects the choice of constraints on reasoning with subject matters.

Names without equality

We can extend our work to the formal language containing both names and variables, $L(\neg, \rightarrow, \forall, P_0, P_1, \ldots, c_0, c_1, \ldots)$.

Now names, as primitive parts of speech, will have subject matter, too. We modify models of **PS** to include:

(10) A set of topics $\mathsf{S} \neq \varnothing$.

An assignment s that for every predicate P, $\mathsf{s}(P) \subseteq \mathsf{S}$ and $\mathsf{s}(P) \neq \varnothing$, and for every name c, $\mathsf{s}(c) \subseteq \mathsf{S}$ and $\mathsf{s}(c) \neq \varnothing$.

$$\mathsf{s}(A) = \bigcup\{\mathsf{s}(P) : P \text{ in } A\} \cup \bigcup\{\mathsf{s}(c) : c \text{ in } A\}$$

Then relatedness is determined by:

$$\mathsf{R}_{\mathsf{s}}(A, B) \text{ iff } \mathsf{s}(A) \cap \mathsf{s}(B) \neq \varnothing$$

It would seem simple to set out semantics for this language. As in the classical case, for each i, for each assignment of references σ, $\sigma(c_i)$ is the same element of the universe. And for any terms $t_1, \ldots, t_n$:

$$\mathsf{v}_\sigma \vDash P_i(t_1, \ldots, t_n) \text{ iff P applied to the sequence } \sigma(t_1), \ldots, \sigma(t_n) \text{ is true.}$$

But this doesn't work, for the following can fail in a model:

(11) $\forall x\, A(x) \rightarrow A(c)$

For example, take $A(x)$ to be $\forall z \neg(P_0(x) \rightarrow P_1(z))$, and a model in which every object in the universe that satisfies the realization of P_0 also satisfies the realization of P_1. If in addition, P_0 and P_1 are unrelated, $\forall x\, A(x)$ is true in the model. But if we choose a name c such that c and P_1 are related, then $\forall z \neg(P_0(c) \rightarrow P_1(z))$ is false in the model.

The reason (11) fails is because as we range over assignments of references in a model:

x_i ranges over individuals referentially
c_i ranges over individuals both referentially and with subject matter

This is not a problem in classical logic, because all a name does is pick out an individual. But even for the classical case, some notion of naming lies behind the recursive definition of truth in a model. As discussed in Chapter IV of Epstein, 1994, the use of an assignment of references is a way to model or abstract from the idea that in evaluating $\forall x\, A(x)$, "no matter what object we pick out and name—perhaps ostensively—that object satisfies A". Indeed, the very notion of a predicate applying to an object depends on abstracting from the idea of naming. It is only with extensional models in which the universe is abstract objects, particularly in applications of classical logic to mathematics, that the notion of naming seems to evaporate. But to take the application of classical logic to mathematics as the archetype does not allow us to consider the relation between formal logic and reasoning in ordinary language.

This way of understanding the evaluation of quantifiers in predicate logic looks much like the substitutional interpretation of quantifiers:

(12) $\forall x\, A(x)$ is true iff for every name t of an object in the universe, $A(t)$ is true

Only here 't' ranges over not just names in the language, but any way we might name objects, including ostensively. When we range over assignments of references in the valuations, we are ranging over ways to name objects of the universe, though these may be abstracted a great deal from naming procedures in our daily lives. What we understand by 'naming' is part of what we specify when we specify a universe.[8]

In the classical case, the only difference between a name in the language and a temporary name that we may give in evaluating a quantified formula is that the name in the language always gets the same element assigned to it. There is nothing more to a name than what it picks out as reference.

Here, there is another difference between a name in the language and a temporary name used in a valuation: A name in the language has some content, namely, subject matter. Temporary names do not have subject matter, and an attempt to give them subject matter is to confuse metalanguage with object language: we could not assign subject matters when we specify the realization. So we adopt the following.

Universal quantifiers and names $v_\sigma(\forall x\, A) = T$
iff for every assignment of references τ that differs from σ at most in what it assigns as reference to x, $v_\tau(A) = T$
and for every name c in the language, $v_\tau(A(c/x)) = T$

This exactly reflects what we said at (12). And this method of evaluating universal quantifiers works equally for classical predicate logic. That is, for any model of classical logic, for every assignment of references σ, for every formula A, $v_\sigma(A) = T$ with the standard evaluation of universal quantifiers, (2), in the model iff $v_\sigma(A) = T$ using the evaluation of universal quantifiers and names.[9]

The logic **PSN** in $L(\neg, \rightarrow, \forall, P_0, P_1, \ldots, c_0, c_1, \ldots)$ is the semantic consequence relation of all **PS** models using (10) for assignments of subject matters and in which the definition of truth in a model uses the evaluation of universal quantifiers and names.

We leave the proof of the next lemma to you.

Lemma 6 *a*. $\forall \ldots \forall x\, A(x) \rightarrow A(c)$ is a scheme of valid formulas in **PSN**.
b. For any model M of **PSN**, $M \models R(A, B)$ iff $R(A, B)$
c. For any model M of **PSN**, for every closed A, for every σ, τ, $v_\sigma \models A$ iff $v_\tau \models A$.
d. The logic **PSN** is contained in classical predicate logic in the language of names without equality.

---

[8] See Chapter IV of Epstein, 1994.

[9] This is the same evaluation of quantifiers discussed and used in the classical free logic presented in Epstein, 2005, though it was first devised for use here.

Lemma 7 **PS** $\subsetneq$ **PSN** restricted to the language without names.

Proof: To show that **PS** $\subseteq$ **PSN**, we show that each axiom of **PS** is true in every **PSN** model. Models of **PSN** preserve truth under applications of *modus ponens* for closed wffs, so each consequence of the axioms is true in **PSN** models, too.

To show that **PS** ≠ **PSN** restricted to the language without names, consider the formula (*) $\exists x \exists z \neg (P_0(x) \rightarrow (P_1(z) \rightarrow P_1(z)))$. This is not valid in **PS**, because we can find a model in which $\mathsf{R}(P_0, P_1)$ fails. But for **PSN**, in every model we have $P_0(c_1) \rightarrow (P_1(c_1) \rightarrow P_1(c_1))$ is true, and hence (*) is, too. ■

The proof of Lemma 7 shows that by allowing our quantifiers to use names as substituends we can create new relatedness where, in the language with no names, there was none before.

We can define the models of **PSN** in terms of relations. Let r be a symmetric, reflexive relation on the collection of all predicates and names in the language. Then define R on all formulas via:

(13) $\mathsf{R}(A, B)$ iff at least one of the following holds:
some P in A, and some Q in B, $\mathsf{r}(P, Q)$
some c in A, and some Q in B, $\mathsf{r}(c, Q)$
some P in A, and some d in B, $\mathsf{r}(P, d)$
some c in A, and some d in B, $\mathsf{r}(c, d)$

We do not know how to characterize (13) in terms of conditions such as R1–R7 for **PS**-relations, and hence how to characterize the relatedness relations syntactically for an axiomatization. We could define $R(P(\vec{x}), Q(\vec{y}))$ as before, but we have no formula A such that $\mathsf{R}(A(c), Q(\vec{y}))$ iff $\mathsf{r}(c, Q)$, since every formula involving c also involves at least one predicate, and that predicate can contribute to the subject matter of A.

In this logic we have continued to assume that predicates are extensional. But since names can be distinguished by their subject matter, we should be able to take into account subject matter when making atomic predications. We could take the application of a predicate to an object to be defined as before, requiring that for all variables x and y, $P(x)$ and $P(y)$ have the same truth-value whenever x and y have the same reference, and similarly for n-ary variables. Then we could make the following definition.

Relatedness application of predicates $P(c)$ is true iff the predicate P applies to the object named by c and $\mathsf{s}(P) \cap \mathsf{s}(c) \neq \varnothing$.

Using this we could have ‘Marilyn Monroe is an actres’ is true in a model while ‘Norma Jean Baker is an actress’ is false.

We do not see how to define this notion within the semantics and syntax of **PSN**, for the same reason that we cannot see how to define in the formal language a formula that corresponds to $\mathsf{R}(P, c)$. Nor do we see how to define the extensional application of predicates from relatedness applications. But we can do so if we introduce equality.

Names and equality

As in our earlier discussion of equality, we have two choices for semantics for the language $L(\neg, \rightarrow, \forall, =, P_1, \ldots, c_0, c_1, \ldots)$.

a. First, consider using the semantics of **PS(=$_u$)** and adding the requirements on subject matters for names from the last section. That is, we add *extensional equality* to the logic **PSN**:

For any names c, d, and any variable x,

$v_\sigma \vDash c = d$ iff $\sigma(c) = \sigma(d)$

$v_\sigma \vDash x = d$ iff $\sigma(x) = \sigma(d)$

$v_\sigma \vDash c = x$ iff $\sigma(c) = \sigma(x)$

For subject matter assignments, we take

A set of topics $S \neq \varnothing$.

An assignment s that for every predicate P, $s(P) \subseteq S$ and $s(P) \neq \varnothing$; and for every name c, $s(c) \subseteq S$ and $s(c) \neq \varnothing$; and $s(\text{'='}) = S$.

$s(A) = \bigcup\{s(P) : P \text{ in } A\} \cup \bigcup\{s(c) : c \text{ in } A\} \cup \{s(\text{'='}) : \text{'=' is in } A\}$

Then relatedness is determined by:

$R_s(A, B)$ iff $s(A) \cap s(B) \neq \varnothing$

The logic **PSN(=$_u$)** is the semantic consequence relation of all such models.

The following fails to be valid in **PSN(=$_u$)**:

$c = d \rightarrow (A(c) \leftrightarrow A(d))$

This is because in a model c may contribute to relatedness within A in a different manner than d does, as in: Marilyn Monroe = Norma Jean Baker $\rightarrow$ [(Marilyn Monroe was an actress $\rightarrow$ an actress married Arthur Miller) $\rightarrow$ (Norma Jean Baker was an actress $\rightarrow$ an actress married Arthur Miller)]' . This reflects that '=' is referential equality, while the evaluation of the conditional takes account of the subject matters of names.

For this logic, too, we cannot see how to define a syntactic equivalent of the relatedness relation of the model, and hence, we cannot see how to axiomatize it.

b. Alternatively, we may conjoin the semantics of **PSN** and **PS(=)**. The definition is exactly as for **PSN(=$_u$)** except that a distinguished element of the set of topics is taken as the subject matter of the equality predicate: $s(\text{'='}) = \{e\}$. The logic of these models is called **PSN(=)**.

In **PSN(=)** the definition $R(A, B) \equiv_{Def} \forall \ldots (A \rightarrow (B \rightarrow B))$ can be used to characterize the relatedness relation of a model syntactically. Then in every model the following holds:

$M \vDash R(x = c, P(y))$ iff $R(c, P)$

And we have that (13) holds as in **PSN(=)**. Using this, we believe that we can axiomatize **PSN(=)**, though we have not done that.

In **PSN(=)** we can define the relatedness application of a predicate. For unary predicates we take:

$$\mathrm{P}_{\mathrm{rel}}(c) \equiv_{\mathrm{Def}} \mathrm{P}(x) \wedge \mathrm{R}(x = c, \mathrm{P}(x))$$

In any model, $\mathsf{M} \vDash \mathrm{P}_{\mathrm{rel}}(c)$ iff $\mathsf{M} \vDash \mathrm{P}(c)$ and $\mathsf{s}(\mathrm{P}) \cap \mathsf{s}(c) \neq \varnothing$.

We can also define a relatedness application of the equality predicate.

$$c =_{\mathrm{rel}} d \equiv_{\mathrm{Def}} c = d \wedge \mathrm{R}(x = c, x = d)$$

Then we have:

$$\mathsf{v}_{\sigma} \vDash c = d \text{ iff } \sigma(c) = \sigma(d) \text{ and } \mathsf{s}(c) \cap \mathsf{s}(d) \neq \varnothing .$$

The abstraction of relatedness models

As for classical predicate logic, we can present an abstraction of models that facilitates a mathematical analysis. We will do this for the logic **PSN**.

For the language $L(\neg, \rightarrow, \forall, \mathrm{P}_1, \ldots, c_0, c_1, \ldots)$ with variables x_0, $x_1, \ldots$, the *abstraction of a relatedness model* is:

$\mathsf{U} \neq \varnothing$ is a set, the *universe*
$\mathsf{S} \neq \varnothing$ is a set, the *topic set*
$\mathsf{S} \cap \mathsf{U} = \varnothing$
Q_i is a subset of *n*-tuples of U for the appropriate *n*,
the *extension* of the predicate assigned to P_i
S_i is a non-$\varnothing$ subset of S,
the (*set-theoretic*) *subject matter* of the predicate assigned to P_i
b_i is an element of U, the *referent* of the name assigned to c_i
s_i is a non-$\varnothing$ subset of S,
the (*set-theoretic*) *subject matter* of the name assigned to c_i

$$\mathsf{M} = \langle \mathsf{U} ; \mathsf{S} ; \langle \mathsf{Q}_0, \mathsf{S}_0 \rangle, \langle \mathsf{Q}_1, \mathsf{S}_1 \rangle, \ldots, \langle \mathsf{b}_0, \mathsf{s}_0 \rangle, \langle \mathsf{b}_1, \mathsf{s}_1 \rangle, \ldots \rangle$$

We write:

P_i for $\langle \mathsf{Q}_i, \mathsf{S}_i \rangle$
a_i for $\langle \mathsf{b}_i, \mathsf{s}_i \rangle$
$\mathsf{s}(\mathsf{P}_i) = \mathsf{S}_i$
$\mathsf{s}(\mathsf{a}_i) = \mathsf{s}_i$

The extension of s to all formulas is via $\mathsf{s}(\mathrm{A}) = \bigcup \{\mathsf{s}(\mathrm{P}) : \mathrm{P} \text{ appears in } \mathrm{A}\}$, the (*set-theoretic*) *subject matter assignment* of M.

The *canonical relatedness relation associated with* s is:

$\mathsf{r}_{\mathsf{s}}(\mathrm{P}_i, \mathrm{P}_j)$ iff $\mathsf{s}(\mathsf{P}_i) \cap \mathsf{s}(\mathsf{P}_j) \neq \varnothing$
$\mathsf{r}_{\mathsf{s}}(c_i, c_j)$ iff $\mathsf{s}(\mathsf{a}_i) \cap \mathsf{s}(\mathsf{a}_j) \neq \varnothing$
$\mathsf{r}_{\mathsf{s}}(c_i, \mathrm{P}_j)$ iff $\mathsf{s}(\mathsf{a}_i) \cap \mathsf{s}(\mathsf{P}_j) \neq \varnothing$
$\mathsf{R}_{\mathsf{s}}(\mathrm{A}, \mathrm{B})$ iff $\mathsf{s}(\mathsf{A}) \cap \mathsf{s}(\mathsf{B}) \neq \varnothing$

An assignment of references σ is any mapping of the variables of the language to elements of the universe, subject to the condition that for any name c, and any two assignments, σ, τ, we have $\sigma(c) = \tau(c)$. The recursive definition of truth in a model uses the extensional application of predicates, the subject matter relatedness evaluation of $\rightarrow$, and the evaluation of universal quantifiers and names:

$v_\sigma(P_i(t_0, \ldots, t_n)) = \mathsf{T}$ iff $<\sigma(t_0), \ldots, \sigma(t_n)> \in \mathsf{Q}_i$

$v_\sigma \vDash t = u$ iff $\sigma(t) = \sigma(u)$

$v_\sigma(\neg A) = \mathsf{T}$ iff $v \nvDash A$

$v_\sigma(A \rightarrow B) = \mathsf{T}$ iff $v_\sigma(A \rightarrow B) = \mathsf{T}$ iff not ($v \vDash A$ and $v \nvDash B$) and $\mathsf{R_s}(A, B)$

$v_\sigma(\forall x\, A) = \mathsf{T}$ iff for every assignment of references τ that differs from σ at most in what it assigns as reference to x, $v_\tau(A) = \mathsf{T}$, and for every name c in the language, $v_\tau(A(c/x)) = \mathsf{T}$

For every closed wff A, $v(A) = \mathsf{T}$ iff for every σ, $v_\sigma(A) = \mathsf{T}$.

We can simplify our notation. Any element in S that does not appear in at least one subject matter assignment contributes nothing to the model and may be ignored. So both S and s can be taken to be given implicitly by the rest of the structure: $\mathsf{S} = \bigcup_i \mathsf{S}_i \cup \bigcup_i \mathsf{s}_i$. Hence, an abstraction of a relatedness model is simply:

$$\mathsf{M} = < \mathsf{U}\ ; <\mathsf{Q}_0, \mathsf{S}_0>, <\mathsf{Q}_1, \mathsf{S}_1>, \ldots, <\mathsf{b}_0, \mathsf{s}_0>, <\mathsf{b}_1, \mathsf{s}_1>, \ldots >$$

We call s, S, and $\mathsf{R_s}$ given implicitly by M above the *associated* s, S, *and* $\mathsf{R_s}$ of M.

Given a relatedness model M, we define the *classical part of* M as:

$$\mathsf{M}_{\text{classical}} = < \mathsf{U}\ ;\ \mathsf{Q}_0,\ \mathsf{Q}_1, \ldots, \mathsf{b}_0,\ \mathsf{b}_1, \ldots >$$

Because we use the extensional application of predicates, any valuation in M and $\mathsf{M}_{\text{classical}}$ will validate the same atomic formulas.

As for the classical case, we can, if we wish, make the *fully general abstraction of relatedness models*:

A model can be comprised of:
any sets $\mathsf{U} \neq \varnothing$, $\mathsf{S} \neq \varnothing$, $\mathsf{S} \cap \mathsf{U} = \varnothing$
and any subsets of n-tuples of U (for appropriate n) $\mathsf{Q}_0, \mathsf{Q}_1, \ldots$
and any elements of U , $\mathsf{b}_0, \mathsf{b}_1, \ldots$,
and any non-empty subsets of S, $\mathsf{S}_0, \mathsf{S}_1, \ldots$, and $\mathsf{s}_0, \mathsf{s}_1, \ldots$.

We leave to you formulations of the abstraction of models for **PS(=$_\mathbf{u}$)**, **PS(=)**, **PSN(=$_\mathbf{u}$)**, and **PSN(=)**.

We believe that this method of giving semantics for logics using set-assignments and relations on propositions can be used to give models for predicate logics for many propositional logics, such as dependence logic, many-valued logics, modal logics, and intuitionistic logic using the set-assignment semantics for those propositional logics as presented in Epstein, 1990.

Bibliography

Epstein, Richard L.

1990 *Propositional Logics* (*The Semantic Foundations of Logic)*
Kluwer. 2nd edition, Oxford University Press, 1995.
2nd edition reprinted with corrections, Wadsworth, 2001.
(All references are to the 2nd ed.)

1994 *Predicate Logic* (*The Semantic Foundations of Logic)*
Oxford University Press. Reprinted Wadsworth, 2001.

2005 *Classical Mathematical Logic* (*The Semantic Foundations of Logic*)
To appear. Princeton University Press.

Grzegorczyk, Andrzej

1974 *An Outline of Mathematical Logic*
D. Reidel Publishing Co.

Krajewski, Stanisław

1986 Relatedness logic
Reports on Mathematical Logic, vol. 20, pp. 7–14.