The Herbrand Manifesto

Thinking Inside the Box

Michael Genesereth and Eric Kao
Computer Science Department
Stanford University

The traditional semantics for Relational Logic (sometimes called Tarskian semantics) is based on the notion of interpretations of constants in terms of objects external to the Logic. Herbrand semantics is an alternative semantics that is based on truth assignments for ground sentences without reference to external objects. Herbrand semantics is simpler and more intuitive than Tarskian semantics; and, consequently, it is easier to teach and learn. Moreover, it is stronger than Tarskian semantics. For example, while it is not possible to finitely axiomatize integer arithmetic with Tarskian semantics, this can be done easily with Herbrand Semantics. The downside is a loss of some common logical properties, such as compactness and inferential completeness. However, there is no loss of inferential power – anything that can be deduced according to Tarskian semantics can also be deduced according to Herbrand semantics. Based on these results, we argue that there is value in using Herbrand semantics for Relational Logic in place of Tarskian semantics. It alleviates many of the current problems with Relational Logic and ultimately may foster a wider use of Relational Logic in human reasoning and computer applications.

Language of Relational Logic

$p(a,b)$ $q(b,c)$

$\neg p(b,d)$ $\forall x.\forall y.(p(x,y) \Rightarrow q(x,y))$

$p(c,b) \vee p(c,d)$ $\exists x.p(x,d)$

One of the main strengths of Relational Logic is that it provides us with a well-defined language for expressing complex information about objects and their relationships. We can write negations, disjunctions, implications, quantified sentences, and so forth.

Deduction in Relational Logic

Premises

$$\forall x.\forall y.(p(x,y) \Rightarrow r(x,y))$$
$$\forall x.\forall y.(q(x,y) \Rightarrow r(x,y))$$
$$\forall x.\exists y.(p(x,y) \vee q(x,y))$$

Conclusion:

$$\forall x.\exists y.r(x,y)$$

Non-Conclusion:

$$\exists y.\forall x.r(x,y)$$

It also provides us with precise rules for deriving conclusions from sentences expressed within this language and a way of knowing which sentences are *not* conclusions.

Semantics

A *model* is a mathematical structure that (directly or indirectly) assigns truth values to sentences in our language.

A set of premises *logically entails* a conclusion *iff* every model that satisfies the premises satisfies the conclusion.

What makes it all work is that the language of Relational Logic has a clearly defined semantics, which gives meaning to logical connectives and quantifiers. This allows us to know that we are using those connectives and quantifiers correctly; and it allows to be sure that, in our reasoning, we are deriving conclusions that follow from our premises and avoiding those that do not.

The basis for almost all treatments of logical semantics is the notion of a model. A model is a mathematical structure that tells us which sentences are true and which are false. And this is the basis for logical entailment. We say that a set of premises logically entails a conclusion if and only if every model that satisfies the premises also satisfies the conclusion. In other words, the conclusion must be true in every model in which the premises are true.

The main question we are addressing today is the nature of these models. In particular, I plan to talk about two alternative types of models - one that is taught in the vast majority of logic courses and one that is used in the majority of work on database and rule systems. They are not the same.

Tarskian Semantics

A vocabulary is a set of symbols.

$$\{a, b, f, q\}$$

A *universe of discourse* is an arbitrary set of objects.

$$\{1, 2, 3, 4, \ldots\}$$

An *interpretation* is an assignment to symbols in language.

$$a=1,\ b=2$$
$$f=\{1\rightarrow 2,\ 2\rightarrow 3,\ \ldots\}$$
$$q=\{\langle 1,2\rangle,\langle 2,3\rangle,\langle 2,4\rangle,\ \ldots\}$$

Entailment defined in terms of all conceivable interpretations

Tarskian semantics is the traditional semantics for Relational Logic. In Tarskian semantics, a model consists of an arbitrary set of objects (called the universe of discourse) and an interpretation function (1) that maps object constants into elements of this set, (2) that maps function constants into functions on this set, and (3) that maps relation constants into relations on this set. A model of this sort completely determines the truth or falsity of all sentences in the language. And it gives us a definition of logical entailment. Note, however, that there are unboundedly many interpretations for any language, and entailment is defined over all conceivable universes - finite, countably infinite, uncountable, and beyond. It also requires an understanding of relations as set of tuples of objects.

Herbrand Semantics

A vocabulary is a set of symbols.

$$\{a, b, p, q\}$$

The Herbrand base is the set of all ground atoms.

$$\{p(a), p(b), q(a,a), q(a,b), q(b,a), q(b,b)\}$$

A *Herbrand model* is any subset of the Herbrand base.

$$\{p(a), q(a,b), q(b,a)\}$$

Entailment defined in terms of symbols only.
Thinking inside the box

Herbrand semantics is somewhat simpler. We start out with the notion of a Herbrand base, simply the set of ground atoms in our language. A model is a simply a subset of the Herbrand base, viz. the elements that are assigned the value true. Or, equivalently, it is a truth assignment for the the elements of the Herbrand base. Given the meaning of logical connectives and quantifiers, a Herbrand model completely determines the truth or falsity of all sentences in the language, not just the ground atoms. And it gives us a definition of logical entailment. One important difference from Tarskian semantics is that Herbrand semantics is less open-ended. There is no external universe, only symbols and sentences in the language. In a sense, it is thinking inside the box.

In much of the literature, Herbrand semantics is treated (somewhat understandably) as a special case of Tarskian semantics - the case where we look at so-called Herbrand interpretations. One downside of this is that it has not been given as much theoretical attention as Tarskian semantics.

Comparison

Benefits of Tarski over Herbrand:
- Compact
- Complete proof procedure
- Semi-decidable

Benefits of Herbrand over Tarski:
- Simpler and more intuitive
- Equivalent or greater inferential power
- More things definable
- Possible unifier of Relational Logic and Rules

In order to understand the trade-off between these two approaches to semantics. Eric Kao and I decided to turn things upside down, focussing Herbrand semantics in its own right instead of as a special case of Tarskian semantics. The results were interesting. On the one hand, we realized quite quickly that we no longer have many of the nice features of Tarskian semantics - compactness, inferential completeness, and semidecidability. On the other hand, we found some real benefits as well. Most importantly, Herbrand semantics is conceptually a lot simpler than Tarskian semantics; as a result, it is easier to teach and learn. It has equivalent or greater inferential power. And more things are definable with Herbrand semantics than with Tarskian semantics. Possible unifier of relational logic and logic programming rules.

My goal today is to go over some of these results with you, as a way of explaining my newfound enthusiasm for Herbrand semantics and my view that it ought to supplant Tarskian semantics in much CS research and most logic education.

Pronunciation

French:

er-brahn

English / American:

her-brand

Stanley Peters:

Mike's ***hare-brained*** logic

Before I get into all of that, a word on pronunciation. Herbrand semantics is named after the logician Jacques Herbrand, who developed some of its key concepts. As Herbrand is French, it should properly be pronounced "air-brahn". However, most people resort to the usual Anglicization of this, instead pronouncing it "her-brand". One exception is Stanley Peters, who frequently to this work as “Mike's hare-brained logic".

Nuts and Bolts

Let's start with the basics - the definition of Herbrand semantics and some of the differences between Herbrand and Tarskiamn semantics.

Herbrand Semantics

The *Herbrand base* is the set of all ground atoms.

$$\{p(a), p(b), q(a,a), q(a,b), q(b,a), q(b,b)\}$$

A *Herbrand model* is an arbitrary subset.

$$\{p(a), q(a,b), q(b,a)\}$$

Equivalently, a *truth assignment* for the ground atoms.

$$\begin{array}{lll} p(a)=1 & q(a,a)=0 & q(b,a)=1 \\ p(b)=0 & q(a,b)=1 & q(b,b)=0 \end{array}$$

Recall that we start out with the notion of a Herbrand base, simply the set of ground atoms in our language. We define a model as an arbitrary subset of this set. Equivalently, it is a truth assignment for the ground atomic sentences in the language.

Truth of Complex Sentences

Ground Atomic Sentences:

$$\varphi = 1 \text{ iff } \varphi \in \Delta$$

Ground Logical Sentences:

$$(\neg\varphi) = 1 \quad \text{iff} \quad \varphi = 0$$
$$(\varphi \wedge \psi) = 1 \quad \text{iff} \quad \varphi = 1 \textit{ and } \psi = 1$$
$$(\varphi \vee \psi) = 1 \quad \text{iff} \quad \varphi = 1 \textit{ or } \psi = 1$$
$$(\varphi \Rightarrow \psi) = 1 \quad \text{iff} \quad \varphi = 0 \textit{ or } \psi = 1$$
$$(\varphi \Leftrightarrow \psi) = 1 \quad \text{iff} \quad \varphi = \psi$$

Quantified Sentences:

$$\forall x.p(x) = 1 \text{ iff } p(\tau) = 1 \text{ for } \textit{all} \text{ ground } \tau$$
$$\exists x.p(x) = 1 \text{ iff } p(\tau) = 1 \text{ for } \textit{some} \text{ ground } \tau$$

Given this notion, the definition of satisfaction is straightforward. A ground atom is true iff it is in the dataset. The truth values of logical sentences are defined the same as with Tarskian semantics. A negation is true iff the negated sentence is false. A conjunction is true iff the conjuncts are both true. And so forth. Finally, a universally quantified sentence is true if and only all of its instances are true.

Herbrand Entailment

A set of premises *logically entails* a conclusion *iff* every model that satisfies the premises satisfies the conclusion.

$$\{\forall x.(p(x) \Rightarrow q(x)),\ p(a)\} \models q(a)$$

Finally, we say that a set of premises logically entails a conclusion if and only if every model that satisfies the premises also satisfies the conclusion. In other words, the conclusion must be true whenever the premises are true. That's the entire semantics.

This is simpler than Tarskian semantics. But it is not entirely equivalent.

Stanford CS Qualifying Exam Question

If $\Delta \models p(\tau)$ for *every* ground term τ, does $\Delta \models \forall x.p(x)$?

Suppose $\Delta \models \{p(a), p(f(a)), p(f(f(a))), p(f(f(f(a)))), ...\}$.
Is it the case that $\Delta \models \forall x.p(x)$?

Let's look at an example. For many years, I was the author of a portion of Stanford's doctoral comprehensive exam. One of my favorite questions concerned logical entailment. Suppose we are given a set Delta of sentences in the language of first-order logic such that Delta logically entails phi(tau) for every ground term in the language. Is it the case that Delta logically entails AX:phi(X)?

Stanford CS Qualifying Exam Question

If $\Delta \models p(\tau)$ for *every* ground term τ, does $\Delta \models \forall x.p(x)$?

Suppose $\Delta \models \{p(a), p(f(a)), p(f(f(a))), p(f(f(f(a)))), \ldots\}$.
Is it the case that $\Delta \models \forall x.p(x)$?

Most Common Answer: *yes*
Correct Answer with Tarskian Semantics: *no*
Correct Answer with Herbrand Semantics: *yes*

The question is not difficult if one knows the semantics of FOL, but apparently not everyone knows this. The most common answer to this question is 'yes' - people seem to think that, if Delta logically entails every ground instance of phi, it must entail the universally quantified version. Of course, that answer is wrong. There can be some unnamed element of the universe of discourse for which the sentence is false.

However, the popularity of the incorrect answer suggests that perhaps our semantics does not capture our intuitions about logic. Maybe it should. The good news is that, with Herbrand semantics, the answer to this question is 'yes' as most people think it should be. Moreover, Herbrand semantics solves not just this problem but a variety of other problems as well.

Compactness

A logic is *compact* if and only if every unsatisfiable set of sentences has a finite subset that is unsatisfiable.

Significance: Possible to demonstrate unsatisfiability in finite space; alternatively, proofs are finite.

Theorem: Logic with Tarskian semantics is compact.

First of all, there is compactness. A logic is compact if and only if every unsatisfiable set of sentences has a finite subset that is unsatisfiable.

Relational Logic with Tarskian semantics turns out to be compact. The upshot is that it is possible to demonstrate unsatisfiability in finite space; alternatively, all proofs are finite.

By contrast, Herbrand Logic is not compact - there are infinite sets of sentences that are unsatisfiable while every finite subset is satisfiable. Consider the set of sentences shown here. It is clearly unsatisfiable under Herbrand semantics; but, if we remove any one sentence, it becomes satisfiable.

The upshot of this is that there are infinite sets of sentences where we cannot demonstrate unsatisfiability with a finite argument within the language itself. Fortunately, this does not cause any practical difficulties, since in all cases of practical interest we are working with finite sets of premises.

Noncompactness

Theorem: Logic with Herbrand semantics is *not* compact.

$$\{p(0), p(s(0)), p(s(s(0))), \ldots, \exists x.\neg p(x)\}$$

Significance: Finite proofs not always possible.

First of all, there is compactness. A logic is compact if and only if every unsatisfiable set of sentences has a finite subset that is unsatisfiable.

Relational Logic with Tarskian semantics turns out to be compact. The upshot is that it is possible to demonstrate unsatisfiability in finite space; alternatively, all proofs are finite.

By contrast, Herbrand Logic is not compact - there are infinite sets of sentences that are unsatisfiable while every finite subset is satisfiable. Consider the set of sentences shown here. It is clearly unsatisfiable under Herbrand semantics; but, if we remove any one sentence, it becomes satisfiable.

The upshot of this is that there are infinite sets of sentences where we cannot demonstrate unsatisfiability with a finite argument within the language itself. Fortunately, this does not cause any practical difficulties, since in all cases of practical interest we are working with finite sets of premises.

Soundness and Completeness

A proof procedure is *sound* if and only if everything that is provable is true.

$$\Delta \vdash \varphi \rightarrow \Delta \models \varphi$$

A proof procedure is *complete* if and only if everything that is true is provable.

$$\Delta \models \varphi \rightarrow \Delta \vdash \varphi$$

More disturbing is that there is no complete proof procedure for Relational Logic with Herbrand semantics. Godel's incompleteness theorem tells us that the set of all true sentences of Peano arithmetic is not computably enumerable. Our axiomatization is complete using Herbrand Semantics. If Herbrand entailment were semi-decidable, the set of all true sentences would be enumerable. Consequently, there is no complete (semi-decidable) proof procedure for Relational Logic with Herbrand semantics.

Tarskian Semantics

There is a proof procedure for relational logic with Tarskian semantics that is both sound and complete.

Moreover, by systematically applying the procedure, possible to compute all logical consequences of any enumerable set of premises.

Hence, provability and logical entailment are semi-decidable.

More disturbing is that there is no complete proof procedure for Relational Logic with Herbrand semantics. Godel's incompleteness theorem tells us that the set of all true sentences of Peano arithmetic is not computably enumerable. Our axiomatization is complete using Herbrand Semantics. If Herbrand entailment were semi-decidable, the set of all true sentences would be enumerable. Consequently, there is no complete (semi-decidable) proof procedure for Relational Logic with Herbrand semantics.

Herbrand Semantics

Axioms of Peano Arithmetic determine the truth of all ground atoms and from that Herbrand semantics tells us the truth of all sentences. The theory is complete in Herbrand semantics.

If Herbrand entailment were semi-decidable, the set of all true sentences would be enumerable.

Godel's incompleteness theorem tells us that the set of all true sentences of Peano arithmetic is not computably enumerable.

Consequently, there is no complete (semi-decidable) proof procedure for Relational Logic with Herbrand semantics.

More disturbing is that there is no complete proof procedure for Relational Logic with Herbrand semantics. Godel's incompleteness theorem tells us that the set of all true sentences of Peano arithmetic is not computably enumerable. Our axiomatization is complete using Herbrand Semantics. If Herbrand entailment were semi-decidable, the set of all true sentences would be enumerable. Consequently, there is no complete (semi-decidable) proof procedure for Relational Logic with Herbrand semantics.

Herbrand Semantics

Theorem: Any sound proof procedure for Tarskian semantics is sound for Herbrand semantics.

The logic is *not* weaker. In fact, it is stronger. There are *more* things that are true. We cannot prove them all, but we can prove everything we could prove before.

By building in induction, we can prove *more* things.

Anything they can do we can do better.

However, this is not as bad as it seems. It turns out that everything that is true under Tarskian semantics is also true under Herbrand semantics, so we can use the same rules of inference. The upshot here is that we lost nothing by switching to Herbrand semantics. In fact, we can add some additional rules of inference. It is not that Herbrand logic is weaker. In fact, it is stronger. There are more things that are true. We cannot prove them all, but we can prove everything we could prove before.

Some may be disturbed by the fact that Herbrand entailment is not semi-decidable. But a similar argument could be leveled against Tarskian semantics. Semi-decidability is not that great either, as a prof procedure might still run forever if the proposed conclusion does not follow from the premises.

Not in Kansas Anymore

So much for standard logical properties. It turns out that, when we look at the use of logic on some practical problems things get more interesting.

Other Interesting Uses

The power of Herbrand semantics is due to the implicit domain closure - there are no objects in the universe except for ground terms.

This allows complete definitions for things that cannot be completely defined with Tarskian Semantics.

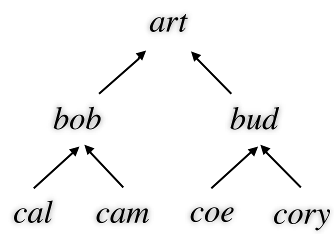
The power of Herbrand semantics is, in large part, due to the implicit property of domain closure – there are no objects in the universe except for ground terms. This allows us to give complete definitions to things that cannot be completely defined with Tarskian Semantics. We have already seen Peano arithmetic. It turns out that, under Herbrand semantics, we can also define some other useful concepts that are not definable with Tarskian semantics, and we can do so without resort to more complex logical mechanisms, such as negation as failure.

Transitive Closure

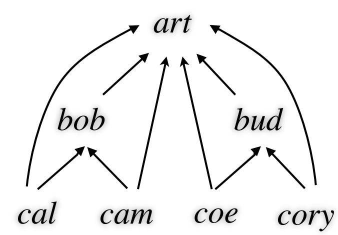
Standard "definition":

$$\forall x.\forall z.(q(x,z) \Leftrightarrow p(x,z) \vee \exists y.(q(x,y) \wedge q(y,z)))$$

Base relation p



Desired closure q



Unfortunately, the desired closure is not the only possibility.

Let's look at transitive closure first. Let's say that we have a binary relation p and we want to axiomatize the transitive closure q.

The typical approach in Relational Logic would be to write the definition shown here.

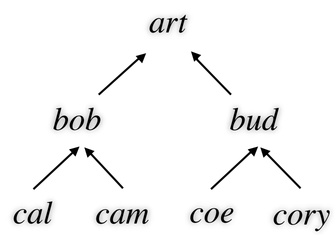
It is easy to see that q contains the transitive closure of p. The problem is that, in general, it can contain additional elements as well, corresponding to various non-standard models. For example, the universe of discourse might contain an object that does not have any p relationships at all. However, if we link all objects to it via q, this satisfies our definition. The upshot is that we have a model of our sentence that is a proper superset of the transitive closure of p. Not good.

Non-Standard Model

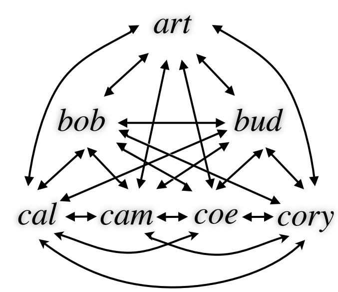
Standard "definition":

$$\forall x.\forall z.(q(x,z) \Leftrightarrow p(x,z) \vee \exists y.(q(x,y) \wedge q(y,z)))$$

Base relation *p*



Non-standard closure *q*



There is no finite set of sentences that defines transitive closure of infinite relations under Tarskian semantics.

The problem is that, in general, it can contain additional elements as well, corresponding to various non-standard models. For example, the universe of discourse might contain an object that does not have any p relationships at all. However, if we link all objects to it via q, this satisfies our definition. The upshot is that we have a model of our sentence that is a proper superset of the transitive closure of p. Not good.

Problem and Solution

What is needed is some form of minimization.

Fortunately, this can be done with Herbrand semantics but not with traditional Tarskian semantics.

Possible to write a definition that works under Herbrand semantics but does not work under Tarskian semantics.

By contrast, we *can* define the transitive closure of a relation in a Logic with Herbrand semantics. It is not as simple or intuitive as this simple definition, but it is theoretically possible. The trick is to exploit the enumerability of the Herbrand universe. Suppose we have the object constant 0, an arbitrary unary relation constant s; and suppose our job is to define q as the transitive closure of p.

We start by defining a helper relation qq as shown below. The basic idea here is that qq(X,Z,N) is the transitive closure in which no intermediate variable is bigger than N. Once we have qq, we can easily define q in terms of qq. q is true of two elements if and only if there is a level at which qq becomes true of those elements.

It is easy to see that q is exactly the transitive closure of p. The only disadvantage of this axiomatization is that we need the helper relation qh. But that causes no significant problems.

Transitive Closure in Herbrand

Definition of helper relation *qq*:

$$\forall x.\forall z.(qq(x,z,0) \Leftrightarrow p(x,z))$$
$$\forall x.\forall z.(qq(x,z,s(n)) \Leftrightarrow qq(x,z,n) \vee \exists y.(qq(x,y,n) \wedge qq(y,z,n)))$$

Definition of *q* in terms of *qq*:

$$\forall x.\forall z.(q(x,z) \Leftrightarrow \exists n.qq(x,z,n))$$

The extra parameter here has the effect of minimizing the number of instances of the q relation.

Theorem: If a model satisfies these axioms under Herbrand semantics, then *q* is the transitive closure of *p*.

By contrast, we *can* define the transitive closure of a relation in a Logic with Herbrand semantics. It is not as simple or intuitive as this simple definition, but it is theoretically possible. The trick is to exploit the enumerability of the Herbrand universe. Suppose we have the object constant 0, an arbitrary unary relation constant s; and suppose our job is to define q as the transitive closure of p.

We start by defining a helper relation qq as shown below. The basic idea here is that qq(X,Z,N) is the transitive closure in which no intermediate variable is bigger than N. Once we have qq, we can easily define q in terms of qq. q is true of two elements if and only if there is a level at which qq becomes true of those elements.

It is easy to see that q is exactly the transitive closure of p. The only disadvantage of this axiomatization is that we need the helper relation qh. But that causes no significant problems.

Curiouser and Curiouser

But wait. There's more! As we know, it is possible to encode some relation in logic programs that cannot be encoded in Relational Logic with Tarskian semantics. Rule systems get this power from the use of negation as failure to minimize defined relations.

Logic Programs

Logic programs are collections of rules.

```
t(0) :-
t(Y) :- t(X) & r(X,Y)
```

A rule is *safe* if and only if every variable in the head occurs in a positive subgoal.

A set of rules is *stratified* if and only if no relation is defined (directly or indirectly) in terms of its own negation.

Safe, stratified logic programs have unique minimal models.

The cool thing is that, even without any form of negation as failure, it is possible to encode those relations in FOL with Herbrand semantics. Moreover, various minimization policies can result from different axiomatizations. We can extend the transitive closure technique to logic programs, showing that the effect of model minimization can be obtained without extensions to the logic.

Transformation

Original Program:

```
t(0) :-
t(Y) :- t(X) & r(X,Y)
```

Normalized Program:

```
t(Y) :- Y=0
t(Y) :- t(X) & r(X,Y)
```

Transformed Program:

$$\forall y.(tt(y,0) \Leftrightarrow y = 0)$$

$$\forall y.\forall n.(tt(y,s(n)) \Leftrightarrow tt(y,n) \vee \exists x.(tt(x,n) \wedge r(x,y)))$$

Projection:

$$\forall y.(t(y) \Leftrightarrow \exists n.tt(y,n))$$

Consider the logic program we just saw. The first step of our conversion is to normalize the program so that the head of every rule consists of variables. This is easy to do using equality (defined as we did earlier in Peano Arithmetic).

Next we transform the normalized program as follows. The two sentences here are the result of transforming the original axioms to define the helper relation ss. For each rule defining an n-ary relation in the normalized program, we define the corresponding n+1-ary helper relation as shown here. ss is true of y and 0 iff y=0. ss is true of y and s(n) iff (1) is true on step n *or* (2) if there is an x such that the definition holds of elements on step n.

Finally, we define s in terms of ss, as we did in the transitive closure example.

Logic Programs

Theorem: Let $\mathcal{P}$ be an arbitrary safe, stratified program over R. Let M be the unique minimal model of $\mathcal{P}$ under stratified semantics. Then, this transformation has a unique model M' under Herbrand semantics such that M' = M over R.

Now the interesting thing is that it turns out that we can do this transformation in general (for arbitrary logic programs so long as they are safe and stratified). And the result is quite good. Let P be an arbitrary safe, stratified program over R. Let M be the unique minimal model of P under stratified semantics. Then, this transformation has a unique model M' under Herbrand semantics such that M' = M over R. Voila - minimization without negation as failure.

Advantages and Disadvantages

Upshot:
:- can be thought of as syntactic sugar
no theoretical need for minimization semantics

Disadvantage:
:- can be used to encode definitions more naturally
computation with :- more efficient

One consequence of this result is that we can treat :- as syntactic sugar for definitions requiring minimization. There is no need for a different logic. Which does note mean that :- is unimportant. In fact, the oddity of those definitions makes clear the value of :- in expressing definitions intuitively.

I think there is also another, more subtle benefit of this theorem. One possible practical consequence of this work concerns the relationship between rule systems and ordinary logic. Rules and ordinary logic are often are often seen as alternatives. Herbrand semantics has the potential to bring these two branches of Logic closer together in a fruitful way. This upshot would be a possible re-prioritization of research in these two areas.

To me, the power and beauty of rule systems is their suitability for writing complete definitions. We start with some completely specified base relations and define other relations in terms of these base relations, working our way up the definitional hierarchy. At every point in time we have a complete model of the world.

Unfortunately, complete theories are not always possible; and in such situations we need to provide for expressing incomplete information. In an attempt to deal with incomplete information, researchers have proposed various extensions to rule systems, e.g. negations, disjunctions, and existentials in the heads of rules, unstratified rules systems, and so forth. Unfortunately, extensions like these mar the beauty of rule systems and ruin their computational properties.

The alternative is to switch to Relational Logic in such situations. Unfortunately, for many, that is not an option. As I have argued, FOL with Tarskian semantics is more complex and weaker than is desirable. There is no negation as failure.

Hybrid Language

Why not have the best of both worlds?

Hybrid logic:
- Complete definitions expressed as rules
- Incomplete information expressed in relational logic

One consequence of this result is that we can treat :- as syntactic sugar for definitions requiring minimization. There is no need for a different logic. Which does note mean that :- is unimportant. In fact, the oddity of those definitions makes clear the value of :- in expressing definitions intuitively.

I think there is also another, more subtle benefit of this theorem. One possible practical consequence of this work concerns the relationship between rule systems and ordinary logic. Rules and ordinary logic are often are often seen as alternatives. Herbrand semantics has the potential to bring these two branches of Logic closer together in a fruitful way. This upshot would be a possible re-prioritization of research in these two areas.

To me, the power and beauty of rule systems is their suitability for writing complete definitions. We start with some completely specified base relations and define other relations in terms of these base relations, working our way up the definitional hierarchy. At every point in time we have a complete model of the world.

Unfortunately, complete theories are not always possible; and in such situations we need to provide for expressing incomplete information. In an attempt to deal with incomplete information, researchers have proposed various extensions to rule systems, e.g. negations, disjunctions, and existentials in the heads of rules, unstratified rules systems, and so forth. Unfortunately, extensions like these mar the beauty of rule systems and ruin their computational properties.

The alternative is to switch to Relational Logic in such situations. Unfortunately, for many, that is not an option. As I have argued, FOL with Tarskian semantics is more complex and weaker than is desirable. There is no negation as failure.

Example

Axioms:

$q(x,y) :- p(x,y)$

$q(x,z) :- q(x,y) \wedge q(y,z)$

$p(a,b) \vee p(b,a)$

$\sim\exists x.q(x,x)$

Questions:

$\forall x.\sim p(x,x)$?

$p(a,b) \Rightarrow \sim p(b,a)$?

My argument is that Herbrand semantics for ordinary logic gives us an ideal middle ground between rules and traditional Relational Logic, allowing us to combine rules with ordinary Logic without losing the benefits that each brings to the table. We can use rules for definitions and ordinary logical operators for constraints.

Here we have an example. q is defined to be the transitive closure of p. We know some disjunctive information about p and we know that q is not reflexive. Taken together we should be able to answer questions like the ones shown here. There cannot be an x such that p(x,x). And we know that at most one of p(ab) and p(b,a) can be true (though we do not know which).

The two can co-exist. Now I do not now whether this is practically possible or not. However, I think it is an idea worthy of study. I cannot imagine that the result would be worse than the alternatives currently in place.

Example

Axioms:

$q(x,y) \text{ :- } p(x,y)$
$q(x,z) \text{ :- } q(x,y) \wedge q(y,z)$

$even(0) \text{ :- }$
$even(s(s(x))) \text{ :- } even(x)$

$\forall x.\forall y.(p(x,y) \wedge even(x) \Rightarrow even(y))$

Questions:

$\forall x.\forall y.(q(x,y) \wedge even(x) \Rightarrow even(y))$?
$\sim p(0,s(s(0)))$?

My argument is that Herbrand semantics for ordinary logic gives us an ideal middle ground between rules and traditional Relational Logic, allowing us to combine rules with ordinary Logic without losing the benefits that each brings to the table. We can use rules for definitions and ordinary logical operators for constraints.

Here we have an example. q is defined to be the transitive closure of p. We know some disjunctive information about p and we know that q is not reflexive. Taken together we should be able to answer questions like the ones shown here. There cannot be an x such that p(x,x). And we know that at most one of p(ab) and p(b,a) can be true (though we do not know which).

The two can co-exist. Now I do not now whether this is practically possible or not. However, I think it is an idea worthy of study. I cannot imagine that the result would be worse than the alternatives currently in place.

Education

These results are encouraging. That said, there is a lot we still do not know, e.g. the relationship to answer set programming, the nature of second order and metalevel logic and Herbrand semantics, and so forth.

The one thing that is clear is that Herbrand semantics is simpler than Tarskian semantics. Whether or not we use it in research and applications remains to be seen. However, I am sure that it has clear value in education.

Logic Education

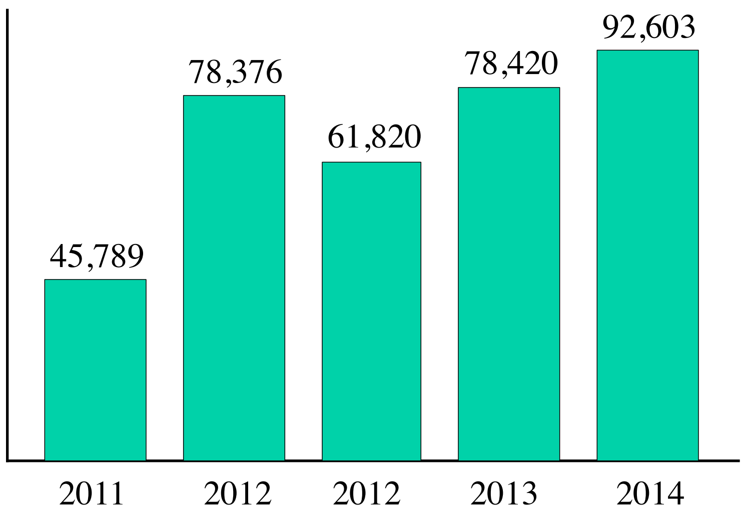
CS157 - Computational Logic course at Stanford
  Introduction to Symbolic Logic
  from computational perspective

Taught for 20 years with Tarskian semantics
  Tarskian Semantics the stumbling block

Taught for 3 years now with Herbrand semantics
  Students do not complain about semantics nearly as much
  Do better on same semantics questions

For many years, while teaching logic the old-fashioned way, I kept thinking that there had to be a better way. I kept thinking that Herbrand semantics might be the answer. To test this, a few years ago, we switched our course on Logic from Tarskian semantics to Herbrand semantics. The results have been gratifying. Students get semantics right away. They do better on quizzes. And there are fewer complaints about feeling lost. From talking with students, I get the sense that many students come away from the course feeling empowered and intent on using Logic. More so than before anyway.

Introduction to Logic MOOC



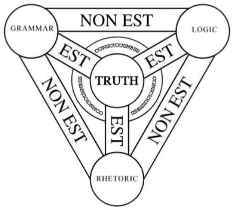
The Logic course is now available a MOOC (and an associated book). It was one of the first MOOCs taught at Stanford. We teach it each year in the Fall. Typical enrollment is now almost 100,000 students per session. To date, almost 500,000 students have enrolled in all. I grant that, as is typical with MOOCs, only a fraction of these students finish. Even so, more students have seen this material than I had taught in my entire life before MOOCs. More students have seen this than have graduated from Stanford's math program in its entire history. So, while mathematicians are still routinely teaching Tarskian semantics in their courses, at least students have an alternative and many are taking advantage of it.

Secondary School

High School
Year long course on Critical Thinking
Logic as 6 week component
Targeting 9th grade

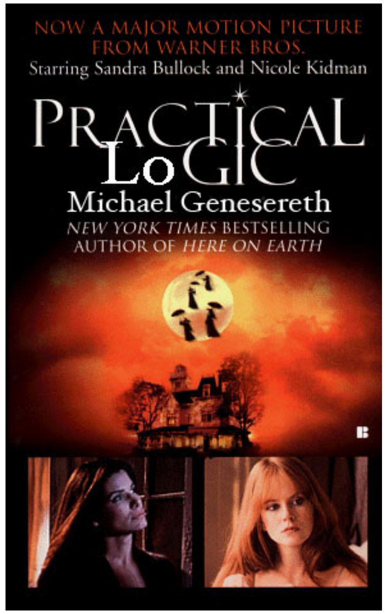
High School
Propositional Logic
Relational Logic with Herbrand Semantics
Logic Programs - Logical Spreadsheets, General Game Playing

Greek Trivium



We are looking at using this approach to teaching logic in secondary schools. In Greek times, Logic was one of the three basic disciplines that students learned. Today, it is taught in only a very small percentage of secondary schools. Instead, we are taught geometry. I learned how to bisect an angle in high school, but no one taught me logic. Since high school I have never really needed to bisect any angles, but I have had to use logic in my professional life and in my private life, e.g. to understand political arguments, court cases, and so forth. But we do not teach logic. I think that if we could make logic more useful and easier to teach, this could change.

Conclusion



During a keynote address at RuleML a few years ago, I talked about Logical Spreadsheets. On that occasion, I mentioned the goal of popularizing logic and suggested that what we need is a way to make logic more accessible and we need tools that make it clear that logic is useful, not just as an intellectual tool but as a practical technology as well.

This time, I have talked about a way to make Logic more accessible – to teach people enough logic so that they can use logical spreadsheets and other logic-based technologies. If Logic easy to learn, my hope is that we can make it more popular. Not just to promote our interests as researchers but also, if we are right, to benefit society with the fruits of our research.

Manifesto being nailed to the door of the Stanford philosophy department.

Universes

Tarskian Semantics

Entailment defined in terms of all conceivable interpretations

Herbrand Semantics

Entailment defined in terms of symbols only

First of all, in Tarskian semantics, there are unboundedly many interpretations for any language, and entailment is defined over all conceivable universes - finite, countably infinite, uncountable, and beyond.

Mathematical Structures

Tarskian Semantics

$$a=1,\ b=2$$
$$f=\{1\rightarrow 2,\ 2\rightarrow 3,\ \ldots\}$$
$$q=\{\langle 1,2\rangle,\langle 2,3\rangle,\langle 2,4\rangle,\ \ldots\}$$

Herbrand Semantics

$$\{p(a),\ q(a,b),\ q(b,a)\}$$

Second, Tarskian semantics requires an understanding of relations as sets of tuples, which is a novel concept for many students. In Herbrand semantics, everything is defined in terms of sentences, which are ore concrete and which students already understand.

Quantified Sentences

Tarskian version:

An interpretation i and a variable assignment s satisfy a universally quantified sentence if and only if i and s satisfy the scope of the sentence for every version of the variable assignment.

A version s[v→x] of a variable assignment s is a variable assignment that assigns v the value x and agrees with s on all other variables.

Herbrand version:

A model satisfies a universally quantified sentence if and only if it satisfies all instances.

Finally, in Tarskian semantics, there is also greater complexity in the definition of satisfaction. Here is the definition in Tarskian semantics. An interpretation i satisfies … A version is … If you got all of that, if you think it is intuitive, then you are better at all of this than I am and my students. Now compare the definition in Herbrand semantics. A dataset satisfies a universally quantified sentences if an only if it satisfies every instance. That's it. Shorter and easier to understand.

Tarskian semantics confuses students. As a result, they feel insecure and are all too often turned off on Logic.