Electronic News Journal on Reasoning about Actions and Change 25
Vol. 3: *25-28*, 31.3.1999
http://www.etaij.org/rac/notes/1999/01/

Guide to Axiomatizing Domains in First-Order Logic

Ernest Davis
Courant Institute, New York University, USA

Ernest Davis has sent us the following guide to the use of first-order logic for formalizing common-sense domains. The guide was written for students that study Knowledge Representation, and it should also be of interest for the ENRAC readership. Comments are welcome and will be organized like other ENRAC discussions.

1. Define a model or microworld, as specifically and precisely as the nature of the domain allows. (Obviously, a much more concrete model can be given in the domain of spatial relations than in the domain of human emotions.) You can then check for the soundness and consistency of your axioms by establishing that they are all true in the model.

   Do not cheat on your model. That is, if you find that you want to include in your axioms some new feature that is not in your model, do not just write down a new axiom. Rather, you should rethink the model to include the new feature, and check that all of your axioms still make sense relative to the revised model.

2. Make two tables enumerating all the sorts and all the non-logical primitives you use in your logic, with their definitions in terms of the model. If it is not possible to give necessary and sufficient conditions for the primitive, write down in English as precise a definition as you can manage, and try at least to bound the concept by giving some necessary conditions and some sufficient conditions.

3. If you want to adopt any notational conventions beyond a simple sorted FOL with equality, write them down explicitly (in English). E.g. if you want to posit the unique names property for constant symbols, write it down. If you have some standard convention for distinguishing fluents from related actions, write it down.

4. Some of the tenets of good programming style hold for axiomatizations as well:

   (a) Non-logical primitives, like programming language identifiers, should be long enough that the reader can immediately know what they mean, but not so long that your axioms must be split up over many lines.

   (b) Modularize. If you find the same subformula coming up again and again, define a new primitive to capture it.

   (c) Use indentation and brackets to make the structure of the axiom apparent.

5. Cast a cold eye on every material implication. Keep in mind that $p \rightarrow q$ means exactly $\neg p \vee q$, and nothing more.

6. If you have written $\forall(X)p(X) \wedge q(X)$ it is likely that you have made a mistake. Rewrite it as the two axioms $\forall(X)p(X)$ and $\forall(X)q(X)$ and see if it still makes sense.

7. If you have written $\exists(X)(p(X) \rightarrow q(X))$ then it is 100 to 1 that you have made a mistake. Rewrite it in the logically equivalent form $(\exists(X)\neg p(X)) \vee (\exists(X)q(X))$ and you will probably realize where you have gone wrong.

   The only exception to this I have ever seen is where the variable X does not actually appear in the left hand side of the implication; for instance, in a form like $\forall(Y)\exists(X)p(Y) \rightarrow q(X,Y)$. Rewrite this for clarity as $\forall(Y)p(Y) \rightarrow \exists(X)q(X,Y)$.

8. If you have the implication $p \rightarrow q$, see if you can turn it into a biconditional $p \leftrightarrow q$. You may have to strengthen the right hand side to $p \leftrightarrow (q \wedge r)$. This can be an easy way to get a more powerful theory.

   This does, however, require some care in choosing a grouping. Forms that are logically equivalent as implications may cease to be so when the implication is turned into a biconditional, and you have to choose the correct form. For instance, the two formulas

   $$\forall(X)(\exists(Y)X = Y + Y) \rightarrow \mathit{even}(X)$$

   and

   $$\forall(X,Y)X = Y + Y \rightarrow \mathit{even}(X)$$

   are equivalent. However, though the first can be turned into the correct biconditional

   $$\forall(X)(\exists(Y)X = Y + Y) \leftrightarrow \mathit{even}(X)$$

   the second cannot.

   Another example: The assertion "If a triangle is equilateral then it is equiangular" can be represented

   $$\forall(X)(\mathit{triangle}(X) \wedge \mathit{equilateral}(X)) \rightarrow$$

   $$\mathit{equiangluar}(X)$$

   To turn this into a biconditional, however, requires separating out the condition *triangle*(X) as a condition on the entire biconditional.

   $$\forall(X)\mathit{triangle}(X) \rightarrow$$
   $$(\mathit{equilateral}(X) \leftrightarrow \mathit{equiangular}(X))$$

9. Equality and function symbols are powerful representational tools; by all means, use them. However, there are a number of common errors to be avoided.

   (a) The formula $X = Y$ means that X and Y are identical things; it is not a translation of *X is Y*. I recently read a paper that contained the formulas, $\mathit{Block1} = \mathit{Red}$; $\mathit{Block2} = \mathit{Blue}$; $\mathit{Block3} = \mathit{Red}$. This would imply that $\mathit{Block1} = \mathit{Block3}$.

   (b) Keep in mind that functions in FOL are necessarily total. Therefore formulas like $\exists(Y)Y = f(X)$ are always true and therefore vacuous.

(c) If you write down an FOL sentence containing the form $X = t$ for some variable X and term t, then you should think whether it is possible to simplify the sentence. In particular, the sentence $\forall(X) X = t \rightarrow p(X)$ is equivalent to $\forall(X) p(t)$. If t does not contain X, then this is equivalent to $p(t)$. Similarly, $\exists(X) X = t \wedge p(X)$ is equivalent to $\exists(X) p(t)$. If t does not contain X, this is equivalent to $p(t)$.

For example, if you are translating the sentence, "Your father is older than you are," (in the sense of "you" meaning "everyone"), it is natural to gloss this as "If X is the father of Y then X is older than Y" and then to write it as

$$\forall(X, Y) X = \mathit{father}(Y) \rightarrow \mathit{older}(X, Y)$$

This can be expressed more elegantly as $\forall(Y) \mathit{older}(\mathit{father}(Y), Y)$.

On the other hand, the original form with the explicit implication may lend itself to conversion to a biconditional, as discussed in (8) above. For instance, the fact, "2 is an even prime," if expressed in the form $\forall(X) X = 2 \rightarrow (\mathit{even}(X) \wedge \mathit{prime}(X))$, can be easily converted to the biconditional "2 is the only even prime," $\forall(X) X = 2 \leftrightarrow (\mathit{even}(X) \wedge \mathit{prime}(X))$.

10. Keep in mind that FOL does not have negation as failure. This is not Prolog. This is particularly important in recursive definitions. If you try to define "above" as the transitive closure of "on" in the Prolog style

$$\forall(X, Y) \mathit{on}(Z, Y) \rightarrow \mathit{above}(X, X)$$
$$\forall(X, Y, Z)(\mathit{on}(X, Y) \wedge \mathit{above}(Y, Z)) \rightarrow \mathit{above}(X, Z)$$

you will be able to prove that some things are above others, but you will never be able to prove that something is not above something else, because this definition is consistent with everything in the world being above everything.

11. Don't do programming. FOL theories describe what's true, not how to compute things. If you find yourself yearning for loops, reassignable variables, data structures, or gotos, then either you're on the wrong track or you should stop working in FOL and switch to a programming language.

12. Check to see whether there exists a simpler or more general model for your axiomatization than the microworld you have in mind. For example, if you think a microworld involves gravity, then imagine a gravity-less version, and ask whether your axioms are still true in that. Another example: some theories of spatial regions in the AI literature are in fact true of arbitrary sets.

Finding such an alternative model can have its pros and cons. Clearly, it means that you have not captured all aspects of your microworld. On the other hand, in a given application, you may not need these aspects. Identifying the axiomatization as a description of a simpler model may lead you to effective algorithms, decidability results, and so on. But, one way or another, it is definitely a fact worth knowing about an axiomatic system.

13. "Seek simplicity and distrust it." — Alfred North Whitehead

Acknowledgements: Thanks to Pat Hayes, Hector Levesque, Vladimir Lifschitz, Rob Miller, and Ray Reiter for helpful criticisms and suggestions. The writing of this note was supported in part by NSF grant #IRI-9625859.