

On My Philosophical Method

by Dennis J. Darland

June 13, 2007

Revised January 16, 2011

Revised (to pdf) May 9, 2011

Copyright © 2011 Dennis J. Darland

No Dogmatism

I am not dogmatic about anything that I say here. I am framing definitions and explanations that I think may solve some philosophical problems or puzzles. I am then trying to see what follows from the definitions and assumptions that I make. The reason for accepting my definitions and assumptions is that they can handle the philosophical problems - because the right conclusions follow. The Preface to Principia Mathematica said something similar about Whitehead and Russell's work:

" ... the chief reason in favour of any theory on the principles of mathematics must always be inductive, i.e. it must lie in the fact that the theory in question enables us to deduce ordinary mathematics. In mathematics, the greatest degree of self-evidence is usually not to be found at the beginning, but at some later point; hence the early deductions, until they reach this point, give reasons rather for believing the premises because true consequences follow from them, than for believing the consequences because they follow from the premises"

Thinking

It seems to me that the educational system tries to make students memorize things, when it should be helping students understand them. If you use something enough you will remember it. There is little point to memorizing something for an exam, only to forget it the next day. If you understand it, even if you don't remember the details, then you will know how to look it up and apply it.

1/15/2011 - There seem to be limits to my previous view. One thing one needs to learn is vocabulary. I now make it a point to look up any words I am not sure of the meaning of - sometimes there are additional meaning to the ones I may already know. But, maybe because I am getting a little old, I find I must look up a word a few times before I remember it.

Testing the Consequences

I've been trying to test that the consequences that I say follows from my definitions really do follow. I have been using Prolog to do that. To help follow my use of Prolog, I will give a very brief explanation of part of Prolog here.

/* A comment. -- First we will state some facts. We will take tom, harry and jack to be people. tom, harry and jack are objects. Objects start with lower case letters. */

/* person and father are predicates. Predicates also start with lower case letters. */

person(tom).

person(harry).

person(jack).

father(tom,harry). /* tom is harry's father - the order is arbitrary but we must settle on one. */

father(jack,tom).

/* We can now define paternal_grandfather. Variables start with a capital letter. */

paternal_grandfather(X,Z) :- father(X,Y), father(Y,Z).

/* I.e. if X is the father of Y and Y is the father of Z then X is the paternal grandfather of Z. */

/* This will be true in Prolog if there are values of variables which satisfy the rule this is checked recursively - I am not showing that here. */

/* We can now make a query. We enter the next line interactively after loading the above facts from one or more files. */

paternal_grandfather(A,B).

/* Prolog will respond something like */

A = jack

B = harry

/* If we enter a semicolon, then Prolog will try to see if there are other cases where it is true - if we just enter a return, then Prolog will stop instead of checking further. */

/* We can also try interactively */

person(tom).

/* Prolog will respond something like: */

Yes

/* but if we try: */

person(bertrand).

/* We get */

No

/* Prolog takes as false anything it cannot prove. This has to be kept in mind. It is more of a problem in real world applications of Prolog than in philosophical applications. We can pretend, in philosophy, that our data is the whole world! */

/* Also, in case you didn't figure it out the predicate left of ':' is defined as true if what is on the right is true. A comma means 'and'. */

/* 'or' is usually gotten by having two rules defining the same predicate. */

Back to [index](#)