

Dennis J. Darland's Philosophy in Prolog

By Dennis J. Darland

June 10, 2007

Revised (to pdf) May 16, 2011

Copyright © 2011 Dennis J. Darland

The Facts

```
/* Prolog database for use with predicates to simulate Dennis J. Darland's philosophy. */  
/* Written 6/10/2007 */  
/* mydb.db */  
belief_r(tom,'denounced','cicero','cataline',now).  
belief_r(tom,'not_denounced','tully','cataline',now).  
symbol1_r(tom,'denounced',denounced,now).  
symbol1_r(tom,'not_denounced',not_denounced,now).  
symbol0_r(tom,'cicero',cicero,now).  
symbol0_r(tom,'cataline',cataline,now).  
symbol0_r(tom,'tully',cicero,now).  
pred(denounced,cicero,cataline).
```

The Predicate definitions

```
/* Prolog predicates to simulate Dennis J. Darland's philosophy. */  
/* Written 6/10/2007 */  
/* myphil.db */
```

understand(S,R,A,B,T) :- symbol1_r(S,W,R,T) ,symbol0_r(S,X,A,T) ,symbol0_r(S,Y,B,T).

logical_form(RF,W,X,Y) :- RF == 'r(a,b)', symbol1_r(S,W,R,T), symbol0_r(S,X,A,T), symbol0_r(S,Y,B,T).

belief(S,R,A,B,T) :- belief_r(S,W,X,Y,T) , symbol1_r(S,W,R,T) ,symbol0_r(S,X,A,T) ,symbol0_r(S,Y,B,T).

proposition(R,A,B) :- understand(S,R,A,B,T).

true_proposition(R,A,B) :- proposition(R,A,B) , pred(R,A,B).

true_belief(S,R,A,B,T) :- belief(S,R,A,B,T) , pred(R,A,B).

symbol0(X) :- symbol0_r(S,X,A,T).

symbol1(X) :- symbol1_r(S,X,A,T).

name_1(W) :- belief_r(S,W,X,Y,T).

name_1(X) :- belief_r(S,W,X,Y,T).

name_1(Y) :- belief_r(S,W,X,Y,T).

Transcript using Above Facts and Definitions

[note the transcript doesn't show the difference between 'cicero' & cicero, but the distinction is made logically in the program.]

Yes

?- [mydb,myphil].

% mydb compiled 0.02 sec, 1,988 bytes

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:5):

Singleton variables: [W, X, Y]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:7):

Singleton variables: [R, A, B]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:11):

Singleton variables: [S, T]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:17):

Singleton variables: [S, A, T]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:19):

Singleton variables: [S, A, T]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:21):

Singleton variables: [S, X, Y, T]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:23):

Singleton variables: [S, W, Y, T]

Warning: (/cygdrive/c/Users/Dennis/src/mine/myphil/myphil.pl:25):

Singleton variables: [S, W, X, T]

% myphil compiled 0.00 sec, 3,076 bytes

Yes

?- name_1(A).

A = denounced ;

A = not_denounced ;

A = cicero ;

A = tully ;

A = cataline ;

A = cataline ;

No

?- symbol1(A).

A = denounced ;

A = not_denounced ;

No

?- symbol0(A).

A = cicero ;

A = cataline ;

A = tully ;

No

?- understand(S,R,A,B,T).

S = tom

R = denounced

A = cicero

B = cicero

T = now ;

S = tom

R = denounced

A = cicero

B = cataline

T = now ;

S = tom

R = denounced

A = cicero

B = cicero

T = now ;

S = tom

R = denounced

A = cataline

B = cicero

T = now ;

S = tom

R = denounced

A = cataline

B = cataline

T = now ;

S = tom

R = denounced

A = cataline

B = cicero

T = now ;

S = tom

R = denounced

A = cicero

B = cicero

T = now ;

S = tom

R = denounced

A = cicero

B = cataline

T = now ;

S = tom

R = denounced

A = cicero

B = cicero

T = now ;

S = tom

R = not_denounced

A = cicero

B = cicero

T = now ;

S = tom

R = not_denounced

A = cicero

B = cataline

T = now ;

S = tom

R = not_denounced

A = cicero

B = cicero

T = now ;

S = tom

R = not_denounced

A = cataline

B = cicero

T = now ;

S = tom

R = not_denounced

A = cataline

B = cataline

T = now ;

S = tom

R = not_denounced

A = cataline

B = cicero

T = now ;

S = tom

R = not_denounced

A = cicero

B = cicero

T = now ;

S = tom

R = not_denounced

A = cicero

B = cataline

T = now ;

S = tom

R = not_denounced

A = cicero

B = cicero

T = now ;

No

?- proposition(R,A,B).

R = denounced

A = cicero

B = cicero ;

R = denounced

A = cicero

B = cataline ;

R = denounced

A = cicero

B = cicero ;

R = denounced

A = cataline

B = cicero ;

R = denounced

A = cataline

B = cataline ;

R = denounced

A = cataline

B = cicero ;

R = denounced

A = cicero

B = cicero ;

R = denounced

A = cicero

B = cataline ;

R = denounced

A = cicero

B = cicero ;

R = not_denounced

A = cicero

B = cicero ;

R = not_denounced

A = cicero

B = cataline ;

R = not_denounced

A = cicero

B = cicero ;

R = not_denounced

A = cataline

B = cicero ;

R = not_denounced

A = cataline

B = cataline ;

R = not_denounced

A = cataline

B = cicero ;

R = not_denounced

A = cicero

B = cicero ;

R = not_denounced

A = cicero

B = cataline ;

R = not_denounced

A = cicero

B = cicero ;

No

?- belief(S,R,A,B,T).

S = tom

R = denounced

A = cicero

B = cataline

T = now ;

S = tom

R = not_denounced

A = cicero

B = cataline

T = now ;

No

?- belief_r(S,R,A,B,T).

S = tom

R = denounced

A = cicero

B = cataline

T = now ;

S = tom

R = not_denounced

A = tully

B = cataline

T = now ;

No

?- true_proposition(R,A,B).

R = denounced

A = cicero

B = cataline ;

R = denounced

A = cicero

B = cataline ;

No

?- true_belief(S,R,A,B,T).

S = tom

R = denounced

A = cicero

B = cataline

T = now ;

No

?- noprotocol.

Back to Top <http://dennisdarland.com/philosophy/index.html>

