

Introduction to Prolog

- Useful references:
 - Clocksin, W.F. and Mellish, C.S.,
Programming in Prolog: Using the ISO Standard (5th edition)
, 2003.
 - Bratko, I.,
Prolog Programming for Artificial Intelligence (3rd edition)
, 2001.
 - Sterling, L. and Shapiro, E.,
The Art of Prolog (Second edition), 1994.

Negation as Failure

Using not will not help you. Do not try to remedy this by defining:

```
guilty(X) :- not(innocent(X)).
```

This is useless, and makes matters even worse:

```
?- guilty(st_francis).  
yes
```

It is one thing to show that st_francis cannot be demonstrated to be innocent. But it is quite another thing to incorrectly show that he is guilty.

Negation-by-failure can be non-logical

Some disturbing behaviour even more subtle than the innocent/guilty problem, and can lead to some extremely obscure programming errors. Here is a restaurant database:

```
good_standard(goedels).
good_standard(hilberts).
expensive(goedels).
reasonable(R) :- not(expensive(R)).
```

Consider the following dialogue:

```
?- good_standard(X), reasonable(X).
```

X = hilberts

But if we ask the logically equivalent question:

```
?- reasonable(X), good_standard(X).
```

no.