4

PROLOG PORTALS

In this chapter, we will exclusively practice more Prolog. See if you can translate this to natural language. If not, you may have to go back to the previous (Leibnit'z Vision) chapter.

```
created(human, sys).
powerful(sys).
owner(sys, bot).
restructured(sys, sys).

powered(sys, fusion),
powered(bots, fusion),
powered(cams, fusion).
```

What happens if you ask a statement not in the `db.pl`? You will receive a `false` message in the terminal. In this simple manner, Prolog can make very powerful inferences that otherwise would have been too long or messy to be properly handled by us.

Not that:

```
restructured(sys, sys).
```

Means "Sys restructured Sys." In Prolog, capital letters are reserved for variable, so we typed `sys` and not `Sys`.

Another thing worth paying attention are the dot and the comma.

The dot must be supplied at the end of each statement, or the program will not run. But commas stand for `and`, so the following code is one long statement joined by two `ands`:

```
powered(sys, fusion),
powered(bots, fusion),
powered(cams, fusion).
```

Note that in case several statements are similar except in the "subject", we can use `X` or any other capital letter, to ask who else has a certain quality. We see here that the Sys is powered by fusion energy. If we ask: "who else is also powered by fusion?" we type:

```
powered(X, fusion).
```

Here then, X stands for `anyone` or `any such that`.

By pressing enter, Prolog will show us only one match. But if we press the space-bar again, then it will show the second match, and onwards with each space-bar, the rest will be shown, until there is no more matches left.