3

LEIBNITZ'S VISION

Leibnitz held the idea that a symbolic language was needed to compute philosophico-logical claims. The reason he envisioned this was his dissatisfaction with the incompleteness, and in some important respects, incorrectness of the Aristotelian logic. For example, we have an example of the four traditional statements:

```
 All bots are rational.

No animals are rational

Some humans are rational.

Some humans are not rational.
```

We encounter a problem when if we make the following inference with the first statement:

```
All bots are rational.

If X is a bot

It is also rational.
```

Syllogistic statements does not have the possibility of possibility; introducing "if" creates problems, and you cannot make propositional claims like "Y exists".

We will continue this discussion, but it will be suitable to start implementing our inferences in a programming language that comes as close as possible to Leibnitz’s idea of mechanistically deriving inferences in order to settle philosophical disputes.

The language we will use is Prolog, more specifically the SWI-Prolog. Prolog is an easy-to-learn programming language that has its roots in first-order logic. To write Prolog, you can download SWI-Prolog, install it on your machine, and then write a database of facts like the one below:

In Prolog:

```
fruit(orange).
eat(cats, fish).
owner(sys, bots).
made(sys, armor, bots).
```

Represent:

Orange is a fruit. Cats eat fish. Sys is the owner of bots. Sys made armor for bots.

Save the database with the .pl extension, like this: "db.pl" where db can be any name. In the terminal, run Swipl and an interactive Prolog session will start.

To import the db.pl file, type:

```
[db].
```

And the `db.pl` file will load. After that, if you type any of the commands in `db.pl`, such as `fruit(orange)`, Prolog will return `true` since what you have asked in the terminal matches what is in `db.pl`.