MARIA KOILALOU *JUSTINE LE BOURG*

Artificial Intelligence

Laboratory 7

Varient 1

Prolog is a logic programming language that operates based on logical reasoning. It utilizes logical facts and rules to represent knowledge and an inference engine to solve logical queries. By searching for variable substitutions, the inference engine attempts to satisfy queries using the unification algorithm.

The code begins by establishing a relationship between each month and the number of days it contains. This is done using the `days_in_month/2` predicate. Each call to this predicate specifies a month (represented by a number) and the corresponding number of days.

Next, we have two predicates: `add_date/2` and `sub_date/2`. These predicates are used to add or subtract a specific number of days from a given date. They utilize string manipulation techniques to extract the day and month from the input date. The extracted substrings are then converted to numbers using the `atom_number/2` function.

Now, let's take a closer look at the `add_date/2` predicate. It first checks if the number of days to add is within the remaining days of the current month, based on the `days_in_month/2` predicate. If the number of days to add does not exceed the remaining days, the date is simply updated by adding the specified number of days to the current day.

However, if the number of days to add exceeds the remaining days, the predicate calculates the remaining days after reaching the end of the current month. It then recursively calls the `add_date/2` predicate by incrementing the month and setting the day to the first day of the next month.

The `sub_date/2` predicate follows a similar logic, but in reverse. It checks if the number of days to subtract is less than the current day. If it is, the date is updated by subtracting the specified number of days from the current day.

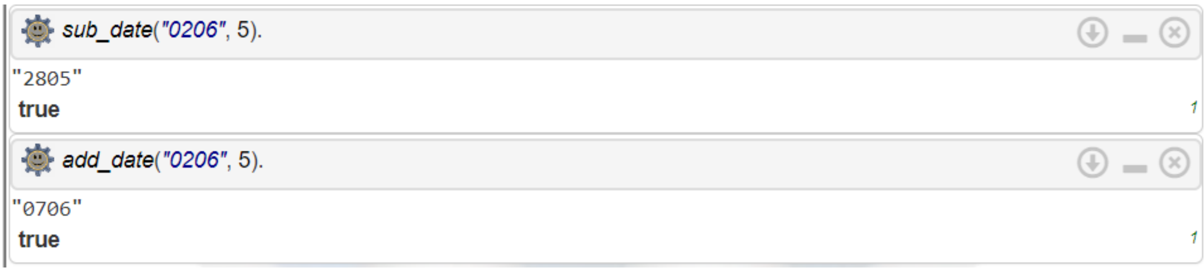
However, if the number of days to subtract exceeds the current day, the predicate calculates the remaining days after reaching the beginning of the current month. It then recursively calls the `sub_date/2` predicate by decrementing the month and setting the day to the last day of the previous month.

Lastly, we have the `format_date/2` predicate, which is used to format the updated date into a string representation in the "DDMM" format. It adds a leading zero to the day or month if necessary and concatenates the strings to form the final date representation.

OUTPUT

To use the code, you have to run it on a website like SWISH -- SWI-Prolog for SHaring, create a program and copy past the code

Then, you can call the `add_date/2` or `sub_date/2` predicates with a date and the number of days you want to add or subtract. The resulting date will be displayed in the "DDMM" format using the `format/2` predicate.



```
sub_date("0206", 5).
"2805"
true
add_date("0206", 5).
"0706"
true
```

These queries add 5 days to the date "0206" (June 2nd) and subtract 5 days from the date "0206", respectively, resulting in the dates "0706" (June 7th) and "2805" (May 28th).