

Tutorial – NLP-1

It is possible to use a pattern matcher to define a simple form of the Elisa program...

```
;; definition of matcher methods for Elisa...

(defmatch elisa (( I like ??a ))
  #?( what do like about ??a -?))

(defmatch elisa (( ??a is ??b ))
  #?( why do you think ??a is ??b -? ))

(defmatch elisa (( ??a are ??b ))
  #?( you think ??a are ??b -? ))

(defmatch elisa (( I am ??a ))
  #?( why do you think you are ??a -? ))

(defmatch elisa (( == ))
  #?( how very interesting ))

;; you can interact with Elisa like this...

> (elisa '(I like Lisp programming))
(what do like about Lisp programming -?)

> (elisa '(Lisp is so much fun))
(why do you think Lisp is so much fun -?)

> (elisa '(because the alternative is VB))
(why do you think because the alternative is VB -?)

> (elisa '(because I get frustrated with other languages))
(how very interesting)
```

Other early (1960s) pattern matching programs (eg: Sir, Student) solved simple problems involving data/storage & retrieval and basic mathematical inference. Pattern matching is not used for processing language now days but experimenting with this approach will help you think about sentence structure and you will quickly experience the limitations of matching.

Use matcher methods to build a program which accepts sentences like...

- (i) a car has 4 wheel
- (ii) a wheel has 5 nut
- (iii) how many nuts does a car have

notes:

- your program should give appropriate responses, ie: report 20 after sentence (iii);
- use only singular nouns because this makes things a bit easier.