CLIPS programming: cheat sheet – v0.1

Manuel J. Marín Jiménez – University of Córdoba (Spain) – mjmarin AT uco

1 Ordered facts

```
(assert (salmorejo tomato bread salt vinegar olive-oil))
(assert (I-have tomato) (I-have olive-oil))
```

2 Non-ordered facts

```
(deftemplate recipe
  (slot name (type SYMBOL) (default ?NONE))
  (multislot ingredients (type SYMBOL) (default ?DERIVE)))
```

```
(assert (recipe (name salmorejo) (ingredients tomato bread
salt vinegar olive-oil)) )
(assert (recipe (name tortilla) (ingredients eggs potato
onion)) )
(modify 1 (name NEW-NAME) )
(duplicate 2 (ingredients NEW-INGREDIENTS) )
(undeftemplate recipe)
```

3 Initial facts

```
(deffacts my-recipes
  (recipe (name salmorejo) (ingredients tomato bread
  salt vinegar olive-oil))
  (recipe (name flamenquin) (ingredients ham pork-loin
  cheese breadcrumbs))
)
(deffacts my-pantry
  (I-have ham)
  (I-have tomato) )
```

```
(list-deffacts)
```

4 Facts: misc

```
(facts) (facts 2 5)
```

```
(retract 1) (retract *)
(reset)
```

5 Rules

```
(defrule find-recipe "Very useful"
   (declare (salience 10))
   (recipe (name ?name) (ingredients $? tomato $?))
  =>
   (printout t "Recipe found: " ?name crlf) )
```

```
(defrule r-union-set
  (or (set1 $? ?v $?)
      (set2 $? ?v $?) )
  (not (union $? ?v $?))
  ?un <- (union $?valores)
 =>
  (retract ?un)
  (assert (union $?valores ?v)) )
```

```
(defrule fire-controlled "All fires controlled?"
    (exists (emergency (type fire)) )
    (forall (emergency (type fire)
                       (location ?where))
            (firemen (location ?where))
    )
    =>
    (printout t "All under control." crlf))
```

```
(defrule you-can-cross
   (logical (traffic-light green))
  =>
   (assert (you-can cross)))
```

6 Rules: misc

```
(agenda)
```

```
(run) (run 3)
(set-strategy random)
```

7 Global variables

```
(defglobal
   ?*gcum* = 0
   ?*ntimes* = 3)
```

```
(bind ?*ntimes* 5)
?*ntimes*
```

8 Modules

```
(defmodule MAIN (export ?ALL))
(defmodule A
  (export ?ALL)
  (import MAIN deftemplate initial-fact recipe) )
(defmodule B (import A ?ALL)
             (import MAIN deftemplate initial-fact) )
```

```
(focus A B)
(return)
```

9 Custom functions

```
(deffunction print-args
  "Show input arguments"
  (?a ?b $?c)
  (printout t ?a ", " ?b " and " (length ?c) " Extra: " $?c crlf))
```

10 Misc

```
(clear)
(exit)
(watch facts) (unwatch facts)
```