**Variables:13**

Programming variables in Jess are identifiers that begin with the question mark (?) character. The question mark is part of the variable's name. The name can contain any combination of letters, numbers, dash (-), underscore(\_), colon (:) or asterisk (\*) characters. Variable names

may *not* contain a period (.) .

A variable can refer to a single symbol, number, or string, or it can refer to a list.

* You can assign a value to to a variable using the bind function:

Jess> (bind ?x "The value")

"The value"

***Dotted variables:13***

New in Jess 7.1 are *dotted variables.* A dotted variable looks like ?x.y. Jess always interprets

this as referring to the slot y of the fact in variable ?x. You can use dotted variables in any

procedural code, but they won't generally work in the pattern matching parts of a rule.

***Global variables (or defglobals):14***

Any variables you create at the Jess> prompt, or at the "top level" of any Jess language

program, are cleared whenever the reset command is issued. This makes them somewhat

transient; they are fine for scratch variables but are not persistent global variables in the normal

sense of the word. To create global variables that are not destroyed by reset, you can use the

defglobal construct.

**Control flow:15**

In Java, control flow -- branching and looping, exception handling, etc -- is handled by special

syntax and keywords like if, while, for, and try. In Jess, as we said before, everything is a

function call, and control flow is no exception. Therefore, Jess includes functions named if,

while, for, and try, along with others like foreach. Each of these functions works similarly to

the Java construct of the same name.

***3.8.1. A simple loop***

For example, a Jess "while" loop looks like this:

**Jess>** (bind ?i 3)

*3*

**Jess>** (while (> ?i 0)

(printout t ?i crlf)

(-- ?i))

*3*

*2*

*1*

*FALSE*

***Functions: 17***

***Working memory:***

***No leer:* Shadow facts: reasoning about Java objects**

**Para el matching de string de los gets, ver p.38, more about regular expresions**