



ARTIFICIAL INTELLIGENCE

LECTURE 8

Ph. D. Lect. Horia Popa Andreescu
2012-2013 3rd year, semester 5

INTRODUCTION

- The content of this lecture is based on the book on expert systems [GR], and on the documentation from Clips website [CLIPS].
- The lecture's topic will be again CLIPS/Jess syntax and programming style.

CONTENTS

- How to write unordered facts (using assert and using deffacts)
- Loading external Jess programs (batch)
- The use of (bind) predicate
- The (modify) predicate
- How to write relational expressions eq, neq, <=, >=, <, >, the (not predicate)
- How to write field constrains e.g
 - (culoare ?x&:(eq ?x rosu))
Will be true if the value of ?x is equal to rosu
 - <http://herzberg.ca.sandia.gov/docs/70/rules.html>
and [GR 382(392)] contain more examples

- Regular expressions (same link, section 6.6)
- (or), (and), (not) (same link, section 6.9-6.11)
- (test) (6.12)
- Logical dependencies among facts (6.13)
- (forall)
- (accumulate)
- (read)

INTERFACING JESS WITH JAVA [6]

Interfacing with Java:

Although it is possible to do almost anything to do with Java within Jess, it is often easier to write a separate Java class and interface with it. The following example calculates the nth fibonacci number with a Java method.

The easy_java.clp file follows:

```
(defglobal ?*external-class* = 0)

(deffunction initialize ()
  (bind ?*external-class* (new easy_java)))

(initialize)

(printout t "The 6th fibonacci number is: " (?*external-class* fibonacci 6) crlf)
```

The easy_java.java file follows

```
public class easy_java{
  public int fibonacci(int n){
    if(n < 3) return 1;
    else return fibonacci(n-1) + fibonacci(n-2);
  }
}
```

INTERFACING JESS WITH JAVA [6]

- To run, compile `easy_java.java`, then run `easy_java.clp` with Jess.
- `Jess> (batch easy_java.clp)`
- *The 6th Fibonacci number is: 8*
-
- Just as in standard Java, an instance of the class must be declared. The first (bind) acts the same as the Java code `easy_java external_class = new easy_java();`. Jess doesn't require periods or parenthesis to call methods, so the call `("?*external-class* fibonacci 6")` is the same as `external_class.fibonacci(6);`.

BIBLIOGRAPHY

- [GR] Giarratano J., Riley G. – Expert Systems Principles and Programming, 3rd edition, PWS Publishing, 2002
- [CLIPS] CLIPS online documentation (visited nov. 2012)
<http://clipsrules.sourceforge.net/OnlineDocs.html>
- [6] Pete Barnum - Basic JESS Tutorial (visited nov. 2012)
<http://www.cs.rochester.edu/~brown/242/docs/JessTutorial.html>
- [RN] Russel S., Norvig P. – Artificial Intelligence – A Modern Approach, 2nd ed. Prentice Hall, 2003 (1112 pages)
- [R] Stuart Russel – Course slides (visited oct. 2012 at
<http://aima.cs.berkeley.edu/instructors.html#homework>)
- [W1] Mark Watson – Practical Artificial Intelligence Programming With Java AI 3rd ed., 2008
- [C] D. Cârstoiu – Sisteme Expert, Editura ALL, București, 1994