**Knowledge Based System Course**

|  |  |
| --- | --- |
| Lab Objective: | * Conflict Resolution in clips |
| Topics: | * Salience * Conflict Resolution Different Strategies |
| Contents | 1. **Salience:** It is an integer number that ranges from (-10000) to (10000) that you give to rules to give them different priorities in execution. The higher the salience, the higher the priority. Default salience for rules if not explicitly defined is zero. 2. **Conflict Resolution in clips:**   **In which order are rules placed on the AGENDA?**   * **SORT by Salience:** Newly activated rules are placed above all rules with lower salience and below all rules with higher salience. * Assume the following rules are on the AGENDA:   Rule-1 salience = 100;  Rule-2 salience = 50;    Rule-3 salience = 75;  Then the order will be as follows:  Rule-1 salience =100;  Rule-3 salience = 75;  Rule-2 salience = 50;  **What if two rules have equal salience?**   * 2. Assume the previous Rule-1 and Rule-2 are on the AGENDA & the newly activated rule has the same salience as Rule-1:   Rule-3 salience = 100 ;  Will the order be?  Rule-1 salience =100;  Rule-3 salience = 100;  Rule-2 salience = 50;  Or?  Rule-3 salience =100;  Rule-1 salience = 100;  Rule-2 salience = 50;  **Based on either one of the following:**   * **Refraction-based:**   + 1. Depth & Breadth * **Specificity-based:**   + 1. Rules with a greater number of conditions are harder to satisfy and are preferred because they take more facts into account.     2. Simplicity and Complexity * **Recency-based:**   + 1. Every fact promoted to the Fact-Base gets a time stamp. This allows us to know which facts are more recent than others.     2. Lex and MEA   **To change the conflict resolution strategy:**   1. (set-strategy depth) 2. (set-strategy breadth) 3. (set-strategy simplicity) 4. (set-strategy complexity) 5. (set-strategy lex) 6. (set-strategy mea) 7. (set-strategy random)   **A Simple Example**   |  |  | | --- | --- | | (defrule rule1         (declare (salience 1))         (f1)         (f2)         (f3) =>         (assert (action1)))  (defrule rule2         (declare (salience 10))         (f2) =>         (assert (action2)))  (defrule rule3         (declare (salience 5))         (f1)         (f2) =>         (assert (action3))) | (defrule rule4         (declare (salience 5))         (f1)         (f2)         (f3) =>         (assert (action4)))  (deffacts the-facts         (f1)         (f2))  (deffacts recent-facts         (f3)) |  * + - 1. **Depth:** * Depth is the default strategy. You only need it to specify it if you have been using another strategy and would like to revert back to it. * How it works: Place the new rules above all rules of same salience:   R1.  R2, R1.  R2, R3, R1.  R2, R4, R3, R1.  **B. Breadth:**   * How it works: Place the new rules below all rules of same salience:   R1.  R2, R1.  R2, R3, R1.  R2, R3, R4, R1.  **C. Simplicity:**   * How it works: Within the same salience class, place newly activated rules above all rules with equal or higher specificity * i.e. priority over rules with more premises to satisfy:   R1  R2, R1  R2, R3, R1  R2, R3, R4, R1  **D. Complexity:**   * How it works: Within the same salience class, place newly activated rules above all rules with equal or lower specificity * i.e. priority over rules of same salience and less premises to satisfy:   R1.  R2, R1.  R2, R3, R1.  R2, R4, R3, R1.  **E. Lex:**   * How it works: Within the same salience class, place newly activated rules above ones with a less recent fact-index * i.e. rules that use more recent facts get priority. In case of a tie, the rule with the higher specificity is placed first:   R1  R2, R1  R2, R3, R1  R2, R4, R3, R1.  **F. MEA:**   * How it works: Within the same salience class, sort rules according to the recency of the data that matches the first pattern. In case of a tie, apply LEX.   R1  R2, R1  R2, R3, R1  R2, R4, R3, R1  **G. Random:**    * Try running the same program more than once with the random strategy. How does it behave? |