**Цель работы** – ознакомление со способами разрешения конфликтов в продукционных системах.

**Постановка задачи:**

Сформировать с помощью конструкции deffacts исходный набор из пяти произвольных фактов (далее обозначаемых (a), (b), (c), (d) и (e)). В соответствии с вариантом задания (табл. 1) сформировать набор правил, где (n), (m), (p), (r), (s) и (t) – некоторые произвольно выбранные факты (в квадратных скобках указана значимость правила).

Таблица 1.

|  |
| --- |
| Вариант 4 |
| (e)(d)(a)=>(p) [8000] (b)(d)=>(m) [8000]  (a)(c)=>(n) [8000]  (a)(d)(b)=>(r) [8000]  (m)(p)=>(t) [6000]  (a)(n)(r)=>(s) [6000] |

**Листинг программы:**

(deffacts f0

(a a)

(b b)

(c c)

(d d)

(e e))

(defrule r01

(declare(salience 8000))

(e e)

(d d)

(a a)

=>

(assert (p p)))

(defrule r02

(declare (salience 8000))

(b b)

(d d)

=>

(assert (m m)))

(defrule r03

(declare (salience 8000))

(a a)

(c c)

=>

(assert (n n)))

(defrule r04

(declare (salience 8000))

(a a)

(d d)

(b b)

=>

(assert (r r)))

(defrule r05

(declare (salience 6000))

(m m)

(p p)

=>

(assert (r r)))

(defrule r06

(declare (salience 6000))

(a a)

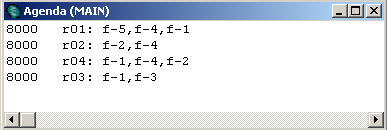
(n n)

(r r)

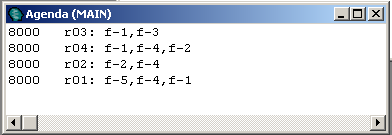
=>

(assert (s s)))

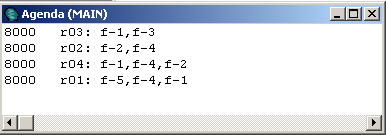
Стратегия глубины:



Стратегия ширины:



Стратегия простоты:



**Вывод**

При стратегии глубины список правил работает подобно стеку, и вновь создаваемые правила помещаются выше всех правил с той же значимостью. При стратегии ширины правила добавляются согласно порядку инициализации фактов. При стратегии простоты правила сортируются согласно простоте с точки зрения сравнений.