Министерство науки и образования РФ

Федеральное государственное бюджетное учреждение

высшего образования

**«Тверской государственный технический университет»**

(ТвГТУ)

Кафедра программного обеспечения

**Отчет по лабораторной работе №6**

по дисциплине: ***«Интеллектуальные информационные системы»***

Тема: ***«Изучение основных возможностей и базовых команд среды CLIPS»***

Выполнил:

Студент группы

Б.ПИН.РИС – 18.06.

Кондратьев М.А.

Проверил:

Мальков А.А.

# Цель

Закрепить понимание принципов решения задач на основе продукционной модели с динамическим изменением подцелей в процессе решения. В процессе работы необходимо реализовать в среде CLIPS программу решения задачи построения башни из блоков.

# Задача

Дан набор разноцветных кубиков разного размера. Расположить все данные кубики в башню сортированными по размеру (снизу самый большой). Вывести ответ в каком порядке расположены кубики. Задача разделена на следующие под цели.

* Найти самый большой кубик данной куче кубиков
* Поставить найденный кубик на верх башни или на пол если это первый кубик
* Если в куче не осталось кубиков, то вывести очередность кубиков

Подготовка

Кубик из себя представляет набор свойств:

* Размер
* Место (heap,tower,hand)
* Цвет (red,yellow,blue,green)

; Шаблон для блока

(deftemplate block

(slot size (type INTEGER) )

(slot place (type SYMBOL ) (default heap) )

(slot color (type SYMBOL ) )

)

Башня из представляется мультислот где блоки расположены по возрастанию

* Блоки

; Шаблон башни  
(deftemplate tower

(multislot blocks (type SYMBOL))

)

Под-цель:

; Шаблон под-цели

(deftemplate goal

(slot task (type SYMBOL))

)

Начальные данные:

; объявляем блоки по варианту

(deffacts var1

(block (size 10) (color blue ))

(block (size 8) (color green ))

(block (size 18) (color red ))

(block (size 15) (color yellow))

(tower)

)

Старт:

; Начальная задача найти наибольший кубик

(defrule start

(initial-fact)

=>

(assert (goal (task find)))

)

Выполниие задачи

Правило поиска самого большого блока в куче и помещения его в руку

(defrule Find

?g <- (goal (task find))

?b <- (block (size ?s1 ) (place heap) )

(not (block (size ?s2 & :(> ?s2 ?s1)) (place heap) ) )

=>

(modify ?b (place hand))

(modify ?g (task build))

)

Правило положить блок в руке наверх башни

(defrule Build

?g <- (goal (task build))

?t <- (tower (blocks $?rest))

?b <- (block (size ?s1) (place hand) (color ?c) )

=>

(modify ?b (place tower))

(modify ?t (blocks ?c ?rest))

(modify ?g (task find))

)

Правило вывода башни пользователю сверху вниз

(defrule Print

?t <- (tower (blocks $?rest))

(test (= (length ?rest) 4))

=>

(progn$ (?c ?rest) (printout t ?c crlf))

)

Пошаговый вывод

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 18) (place heap) (color red))

(block (size 15) (place heap) (color yellow))

(tower (blocks))

start: f-0

----------

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 18) (place heap) (color red))

(block (size 15) (place heap) (color yellow))

(tower (blocks))

(goal (task find))

Find: f-6,f-3,

----------

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 15) (place heap) (color yellow))

(tower (blocks))

(block (size 18) (place hand) (color red))

(goal (task build))

Build: f-8,f-5,f-7

----------

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 15) (place heap) (color yellow))

(block (size 18) (place tower) (color red))

(tower (blocks red))

(goal (task find))

Find: f-11,f-4,

----------

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 18) (place tower) (color red))

(tower (blocks red))

(block (size 15) (place hand) (color yellow))

(goal (task build))

Build: f-13,f-10,f-12

----------

(i-f)

(block (size 10) (place heap) (color blue))

(block (size 8) (place heap) (color green))

(block (size 18) (place tower) (color red))

(block (size 15) (place tower) (color yellow))

(tower (blocks yellow red))

(goal (task find))

Find: f-16,f-1,

----------

(i-f)

(block (size 8) (place heap) (color green))

(block (size 18) (place tower) (color red))

(block (size 15) (place tower) (color yellow))

(tower (blocks yellow red))

(block (size 10) (place hand) (color blue))

(goal (task build))

Build: f-18,f-15,f-17

----------

(i-f)

(block (size 8) (place heap) (color green))

(block (size 18) (place tower) (color red))

(block (size 15) (place tower) (color yellow))

(block (size 10) (place tower) (color blue))

(tower (blocks blue yellow red))

(goal (task find))

Find: f-21,f-2,

----------

(i-f)

(block (size 18) (place tower) (color red))

(block (size 15) (place tower) (color yellow))

(block (size 10) (place tower) (color blue))

(tower (blocks blue yellow red))

(block (size 8) (place hand) (color green))

(goal (task build))

Build: f-23,f-20,f-22

----------

(i-f)

(block (size 18) (place tower) (color red))

(block (size 15) (place tower) (color yellow))

(block (size 10) (place tower) (color blue))

(block (size 8) (place tower) (color green))

(tower (blocks green blue yellow red))

(goal (task find))

Print: f-25

green

blue

yellow

red

# Заключение

Было закреплено понимание принципов решения задач на основе продукционной модели и динамическим изменением подцелей в процессе решения.