МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение высшего образования

«Тверской государственный технический университет»

(ТвГТУ)

Кафедра «Программного обеспечения»

**Отчёт по лабораторной работе №6**

**Вариант 9**

по дисциплине “Системы искусственного интеллекта”

Выполнил: студент группы

ПИН-17.06

Иванов Р. В.

Проверил:

Мальков А. А.

Тверь 2021

**Цель работы** – 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.

**Постановка задачи:**

Задача построения башни из блоков (кубиков) заключается в последовательном выборе из неупорядоченной кучи блоков и постановки их друг на друга. Процесс решения задачи представляет собой чередование двух фаз: выбора блока из кучи и установки его в башню. При этом б*о*льший по размерам блок не может ставиться на м*е*ньший по размерам и, следовательно, на каждом шаге решения задачи необходимо выбирать из кучи самый большой блок.

**Исходные данные (цвет/размер):**

черный/18

желтый/15

крас/20

белый/10

**Листинг:**

(defglobal ?\*in-heap\* = 4) ; задаем размер кучи

(deftemplate block ; задаем шаблон блока

(slot color (type SYMBOL))

(slot size (type INTEGER))

(slot place (type SYMBOL))

)

(deftemplate goal

(slot found (type SYMBOL))

)

(deftemplate on-block

(slot color (type SYMBOL))

(slot up-block (type SYMBOL))

(slot down-block (type SYMBOL))

)

(deftemplate task ; факт, определяющий текущую задачу.

(slot current (type SYMBOL)(allowed-symbols find build))

)

; задаем начальные значения

(deffacts init

(block

(color black)

(size 18)

(place heap)

)

(block

(color yellow)

(size 15)

(place heap)

)

(block

(color red)

(size 20)

(place heap)

)

(block

(color white)

(size 10)

(place heap)

)

)

(defrule init-r ; правило установки начального значения текущей задачи

(initial-fact)

=>

(assert

(task

(current find)

)

)

)

(defrule find-biggest ; Правило выбора из кучи самого большого блока

?tf<-(task

(current find)

)

(test (> ?\*in-heap\* 0))

?pbl<-(block

(size ?sz0)

(place heap)

)

(not (exists (block (place heap)(size ?sz1&:(> ?sz1 ?sz0))) ) )

=>

(modify ?pbl

(place hand)

)

(bind ?\*in-heap\* (- ?\*in-heap\* 1))

(modify ?tf

(current build)

)

)

;----------------------------------------------

(defrule build-first ; Правило установки блока в основание башни

?tf<-( task

(current build)

)

?fbl<-(block

(place hand)

(color ?cl)

)

(not (exists (on-block (up-block undefined))))

=>

(assert (on-block

(color ?cl)

(down-block undefined)

(up-block undefined)

)

)

(modify ?tf

(current find)

)

(modify ?fbl

(place tower)

)

)

;--------------------------------------------

(defrule build-next ; Правило установки в башню последующих блоков

?tf<-( task

(current build)

)

?fbl<-(block

(place hand)

(color ?cl)

)

?onf<-(on-block

(up-block undefined)

(color ?cl-old)

)

=>

(assert (on-block

(color ?cl)

(down-block ?cl-old)

(up-block undefined)

)

)

(modify ?tf

(current find)

)

(modify ?fbl

(place tower)

)

(modify ?onf

(up-block ?cl)

)

)

;-----------------------------------------

(defrule goal-test

(not (exists (block (place heap))))

=>

(printout t "Built! Print from top to bottom" crlf)

(assert (goal (found done)))

)

;-----------------------------------------

(defrule print-tower

(exists (goal (found done)))

?bl<-(on-block

(color ?cl)

(up-block undefined)

(down-block ?cl-down)

)

?bl-under<-(on-block

(color ?cl-down)

)

=>

(printout t "Block: "?cl crlf)

(retract ?bl)

(modify ?bl-under

(up-block undefined)

)

)

;-----------------------------------------

(defrule print-last

?bl<-(on-block

(up-block undefined)

(down-block undefined)

(color ?cl)

)

=>

(printout t "Block: "?cl crlf)

(retract ?bl)

)

**Результат выполнения программы:**

