Лабораторная работа 2

Давитян Эдуард

Нпи-01-21

Отчет:

**Домен:**

(define

(domain coins)

(:types

robot - object

number room - object

container - object

key - object

vendor - object

)

; un-comment following line if constants are needed

(:constants

n0 - number

n1 - number

n2 - number

n3 - number

n4 - number

n5 - number

n6 - number

n7 - number

n8 - number

n9 - number

n10 - number

)

(:predicates ;

(adds ?x - number ?y - number ?z - number); chislo x plus chislo y =z

(adjacent ?x - room ?y - room); mojnoproiti iz komnatu x v komantu y //to est est put

(room\_floor\_coins ?x - room ?y - number); v komnate xnaxoditsa y monet na polu

(robot\_coins ?r - robot ?x - number); u robota R est x monet

(robot\_at ?r - robot ?x - room);R robot naxoditsa v komnate x

(vendor\_is\_at ?v - vendor ?x - room) ;prodavec naxoditsa v x komnate

(vendor\_sells\_key ?v - vendor ?k - key ?n - number);prodavec V prodaet kluch K za N monet

(robot\_has\_key ?r - robot ?k - key); robot R imeet kluch k

(key\_unlocks\_door ?k - key ?x - room ?y - room); kluchotkrivaet put mejdu dvumia komnatami x /y

(room\_has\_container ?x - room ?c - container);konteinernaxoditsa v komnate

(container\_has\_coins ?c - container ?n - number); moneti naxodiatsa v konteinere

)

; esli robot R v current room i current room riadom s next room torobotbudet v next room

(:action move

:parameters (

?r - robot

?current\_room - room

?next\_room - room

)

:precondition (and

(robot\_at ?r ?current\_room)

(adjacent ?current\_room ?next\_room)

)

:effect (and

(not (robot\_at ?r ?current\_room))

(robot\_at ?r ?next\_room)

)

)

; esli na polu est moneti tomi podbiraem kakoeto kolichestvo

(:action pick\_up\_coins\_from\_floor

:parameters (

?r - robot

?current\_room - room

?room\_floor\_coins\_before - number

?room\_floor\_coins\_after - number

?coins\_picked\_up - number

?robot\_coins\_before - number

?robot\_coins\_after - number

)

:precondition (and

(robot\_at ?r ?current\_room)

(room\_floor\_coins ?current\_room ?room\_floor\_coins\_before)

(robot\_coins ?r ?robot\_coins\_before)

(adds ?robot\_coins\_before ?coins\_picked\_up ?robot\_coins\_after)

(adds ?room\_floor\_coins\_after ?coins\_picked\_up ?room\_floor\_coins\_before)

)

:effect (and

(not (room\_floor\_coins ?current\_room ?room\_floor\_coins\_before))

(room\_floor\_coins ?current\_room ?room\_floor\_coins\_after)

(not (robot\_coins ?r ?robot\_coins\_before))

(robot\_coins ?r ?robot\_coins\_after)

)

)

; deistvie kotoroe ukazivaet na tochto robotzabiraet moneti s konteinera

(:action withdraw\_from\_container

:parameters (

?r - robot

?current\_room - room

?container - container

?container\_coins\_before - number

?container\_coins\_after - number

?coins\_withdrawn - number

?robot\_coins\_before - number

?robot\_coins\_after - number

)

:precondition (and

(robot\_at ?r ?current\_room)

(room\_has\_container ?current\_room ?container)

(container\_has\_coins ?container ?container\_coins\_before)

(robot\_coins ?r ?robot\_coins\_before)

(adds ?robot\_coins\_before ?coins\_withdrawn ?robot\_coins\_after)

(adds ?container\_coins\_after ?coins\_withdrawn ?container\_coins\_before)

)

:effect (and

(not (container\_has\_coins ?container ?container\_coins\_before))

(not (robot\_coins?r ?robot\_coins\_before))

(container\_has\_coins ?container ?container\_coins\_after)

(robot\_coins ?r ?robot\_coins\_after)

)

)

;zdes robot kladet moneti v konteiner

(:action deposit\_into\_container

:parameters (

?r - robot

?current\_room - room

?container - container

?container\_coins\_before - number

?container\_coins\_after - number

?coins\_withdrawn - number

?robot\_coins\_before - number

?robot\_coins\_after - number

)

:precondition (and

(robot\_at ?r ?current\_room)

(room\_has\_container ?current\_room ?container)

(container\_has\_coins ?container ?container\_coins\_before)

(robot\_coins ?r ?robot\_coins\_before)

(adds ?robot\_coins\_after ?coins\_withdrawn ?robot\_coins\_before)

(adds ?container\_coins\_before ?coins\_withdrawn ?container\_coins\_after)

)

:effect (and

(not (container\_has\_coins ?container ?container\_coins\_before))

(not (robot\_coins ?r ?robot\_coins\_before))

(container\_has\_coins ?container ?container\_coins\_after)

(robot\_coins ?r ?robot\_coins\_after)

)

)

;kogda robot pokupaet kluch u prodavca

(:action buy\_key

:parameters (

?r - robot

?current\_room - room

?vendor - vendor

?key - key

?key\_price - number

?robot\_coins\_before - number

?robot\_coins\_after - number

)

:precondition (and

(robot\_at ?r ?current\_room)

(vendor\_is\_at ?vendor ?current\_room)

(vendor\_sells\_key ?vendor ?key ?key\_price)

(robot\_coins ?r ?robot\_coins\_before)

(adds ?robot\_coins\_after ?key\_price ?robot\_coins\_before)

)

:effect (and

(not (robot\_coins ?r ?robot\_coins\_before))

(robot\_coins ?r ?robot\_coins\_after)

(robot\_has\_key ?r ?key)

)

)

;etokogda robot otkrivaet dver kluchom

(:action unlock\_door\_from\_left

:parameters (

?r - robot

?current\_room - room

?next\_room - room

?key - key

)

:precondition (and

(robot\_at ?r ?current\_room)

(key\_unlocks\_door ?key ?current\_room ?next\_room)

(robot\_has\_key ?r ?key)

)

:effect (and

(adjacent ?current\_room ?next\_room)

(adjacent ?next\_room ?current\_room)

)

)

;

(:action unlock\_door\_from\_right

:parameters (

?r - robot

?current\_room - room

?next\_room - room

?key - key

)

:precondition (and

(robot\_at ?r ?current\_room)

(key\_unlocks\_door ?key ?next\_room ?current\_room)

(robot\_has\_key ?r ?key)

)

:effect (and

(adjacent ?current\_room ?next\_room)

(adjacent ?next\_room ?current\_room)

)

)

)

**PROBLEM1:**

Идея алгоритма: для каждого i от 0 до 10, для каждого j от 0 до 10, если i+j<10 то (Adds n{i} n{j} n{i+j}), иначе (Adds n{i} n{j} n10).

У нас есть 5 комнат и 5 контейнеров . Робот Робби , ключи и продавец . Из начально у Робби 0 монет , но он должен их собрать , в каждом контейнере по 2 монеты , а в каждой комнате по 1 контейнеру

(define (problem coins1)

(:domain coins)

(:objects

r1 - room ; komanti

r2 - room

r3 - room

r4 - room

r5 - room

c1 - container

c2 - container

c3 - container

c4 - container

c5 - container

robbie - robot

k\_null - key

v\_null - vendor

)

(:init

;pravilo slojenie chisel

(adds n0 n0 n0)

(adds n0 n1 n1)

(adds n0 n2 n2)

(adds n0 n3 n3)

(adds n0 n4 n4)

(adds n0 n5 n5)

(adds n0 n6 n6)

(adds n0 n7 n7)

(adds n0 n8 n8)

(adds n0 n9 n9)

(adds n0 n10 n10)

(adds n1 n0 n1)

(adds n1 n1 n2)

(adds n1 n2 n3)

(adds n1 n3 n4)

(adds n1 n4 n5)

(adds n1 n5 n6)

(adds n1 n6 n7)

(adds n1 n7 n8)

(adds n1 n8 n9)

(adds n1 n9 n10)

(adds n1 n10 n10)

(adds n2 n0 n2)

(adds n2 n1 n3)

(adds n2 n2 n4)

(adds n2 n3 n5)

(adds n2 n4 n6)

(adds n2 n5 n7)

(adds n2 n6 n8)

(adds n2 n7 n9)

(adds n2 n8 n10)

(adds n2 n9 n10)

(adds n2 n10 n10)

(adds n3 n0 n3)

(adds n3 n1 n4)

(adds n3 n2 n5)

(adds n3 n3 n6)

(adds n3 n4 n7)

(adds n3 n5 n8)

(adds n3 n6 n9)

(adds n3 n7 n10)

(adds n3 n8 n10)

(adds n3 n9 n10)

(adds n3 n10 n10)

(adds n4 n0 n4)

(adds n4 n1 n5)

(adds n4 n2 n6)

(adds n4 n3 n7)

(adds n4 n4 n8)

(adds n4 n5 n9)

(adds n4 n6 n10)

(adds n4 n7 n10)

(adds n4 n8 n10)

(adds n4 n9 n10)

(adds n4 n10 n10)

(adds n5 n0 n5)

(adds n5 n1 n6)

(adds n5 n2 n7)

(adds n5 n3 n8)

(adds n5 n4 n9)

(adds n5 n5 n10)

(adds n5 n6 n10)

(adds n5 n7 n10)

(adds n5 n8 n10)

(adds n5 n9 n10)

(adds n5 n10 n10)

(adds n6 n0 n6)

(adds n6 n1 n7)

(adds n6 n2 n8)

(adds n6 n3 n9)

(adds n6 n4 n10)

(adds n6 n5 n10)

(adds n6 n6 n10)

(adds n6 n7 n10)

(adds n6 n8 n10)

(adds n6 n9 n10)

(adds n6 n10 n10)

(adds n7 n0 n7)

(adds n7 n1 n8)

(adds n7 n2 n9)

(adds n7 n3 n10)

(adds n7 n4 n10)

(adds n7 n5 n10)

(adds n7 n6 n10)

(adds n7 n7 n10)

(adds n7 n8 n10)

(adds n7 n9 n10)

(adds n7 n10 n10)

(adds n8 n0 n8)

(adds n8 n1 n9)

(adds n8 n2 n10)

(adds n8 n3 n10)

(adds n8 n4 n10)

(adds n8 n5 n10)

(adds n8 n6 n10)

(adds n8 n7 n10)

(adds n8 n8 n10)

(adds n8 n9 n10)

(adds n8 n10 n10)

(adds n9 n0 n9)

(adds n9 n1 n10)

(adds n9 n2 n10)

(adds n9 n3 n10)

(adds n9 n4 n10)

(adds n9 n5 n10)

(adds n9 n6 n10)

(adds n9 n7 n10)

(adds n9 n8 n10)

(adds n9 n9 n10)

(adds n9 n10 n10)

(adds n10 n0 n10)

(adds n10 n1 n10)

(adds n10 n2 n10)

(adds n10 n3 n10)

(adds n10 n4 n10)

(adds n10 n5 n10)

(adds n10 n6 n10)

(adds n10 n7 n10)

(adds n10 n8 n10)

(adds n10 n9 n10)

(adds n10 n10 n10)

(adjacent r1 r2) (adjacent r2 r1)

(adjacent r2 r3) (adjacent r3 r2)

(adjacent r3 r4) (adjacent r4 r3)

(adjacent r4 r5) (adjacent r5 r4)

;govorit o tom gde naxodiatsa konteineri

(room\_has\_container r1 c1)

(room\_has\_container r2 c2)

(room\_has\_container r3 c3)

(room\_has\_container r4 c4)

(room\_has\_container r5 c5)

;skolko monet vkajdoi komnate na polu

(room\_floor\_coins r1 n3)

(room\_floor\_coins r2 n1)

(room\_floor\_coins r3 n0)

(room\_floor\_coins r4 n4)

(room\_floor\_coins r5 n2)

;konteineri s monetami

(container\_has\_coins c1 n0)

(container\_has\_coins c2 n0)

(container\_has\_coins c3 n1)

(container\_has\_coins c4 n0)

(container\_has\_coins c5 n0)

(robot\_coins robbie n0); s nachalo 0 monet

(robot\_at robbie r1)

)

(:goal (and

(container\_has\_coins c1 n2) ; v kajdom konteinere po 2 moneti

(container\_has\_coins c2 n2)

(container\_has\_coins c3 n2)

(container\_has\_coins c4 n2)

(container\_has\_coins c5 n2)

))

)

**PROBLEM 2**

У нас 5 комнат которые расположены : одна на юге , вторая на востоке , третья на западе , четвертая на севере и есть одна другая комната . Также Робот Робби , и три ключа и продавец. Продавец находится в комнате Rc . В контейнере должно быть 6 монет

(define (problem coins1)

(:domain coins)

(:objects

rn - room

rc - room

rs - room

re - room

ree - room

cn - container

cs - container

cee - container

robbie - robot

kn - key

ks - key

ke - key

vendor - vendor

)

(:init

(adds n0 n0 n0)

(adds n0 n1 n1)

(adds n0 n2 n2)

(adds n0 n3 n3)

(adds n0 n4 n4)

(adds n0 n5 n5)

(adds n0 n6 n6)

(adds n0 n7 n7)

(adds n0 n8 n8)

(adds n0 n9 n9)

(adds n0 n10 n10)

(adds n1 n0 n1)

(adds n1 n1 n2)

(adds n1 n2 n3)

(adds n1 n3 n4)

(adds n1 n4 n5)

(adds n1 n5 n6)

(adds n1 n6 n7)

(adds n1 n7 n8)

(adds n1 n8 n9)

(adds n1 n9 n10)

(adds n1 n10 n10)

(adds n2 n0 n2)

(adds n2 n1 n3)

(adds n2 n2 n4)

(adds n2 n3 n5)

(adds n2 n4 n6)

(adds n2 n5 n7)

(adds n2 n6 n8)

(adds n2 n7 n9)

(adds n2 n8 n10)

(adds n2 n9 n10)

(adds n2 n10 n10)

(adds n3 n0 n3)

(adds n3 n1 n4)

(adds n3 n2 n5)

(adds n3 n3 n6)

(adds n3 n4 n7)

(adds n3 n5 n8)

(adds n3 n6 n9)

(adds n3 n7 n10)

(adds n3 n8 n10)

(adds n3 n9 n10)

(adds n3 n10 n10)

(adds n4 n0 n4)

(adds n4 n1 n5)

(adds n4 n2 n6)

(adds n4 n3 n7)

(adds n4 n4 n8)

(adds n4 n5 n9)

(adds n4 n6 n10)

(adds n4 n7 n10)

(adds n4 n8 n10)

(adds n4 n9 n10)

(adds n4 n10 n10)

(adds n5 n0 n5)

(adds n5 n1 n6)

(adds n5 n2 n7)

(adds n5 n3 n8)

(adds n5 n4 n9)

(adds n5 n5 n10)

(adds n5 n6 n10)

(adds n5 n7 n10)

(adds n5 n8 n10)

(adds n5 n9 n10)

(adds n5 n10 n10)

(adds n6 n0 n6)

(adds n6 n1 n7)

(adds n6 n2 n8)

(adds n6 n3 n9)

(adds n6 n4 n10)

(adds n6 n5 n10)

(adds n6 n6 n10)

(adds n6 n7 n10)

(adds n6 n8 n10)

(adds n6 n9 n10)

(adds n6 n10 n10)

(adds n7 n0 n7)

(adds n7 n1 n8)

(adds n7 n2 n9)

(adds n7 n3 n10)

(adds n7 n4 n10)

(adds n7 n5 n10)

(adds n7 n6 n10)

(adds n7 n7 n10)

(adds n7 n8 n10)

(adds n7 n9 n10)

(adds n7 n10 n10)

(adds n8 n0 n8)

(adds n8 n1 n9)

(adds n8 n2 n10)

(adds n8 n3 n10)

(adds n8 n4 n10)

(adds n8 n5 n10)

(adds n8 n6 n10)

(adds n8 n7 n10)

(adds n8 n8 n10)

(adds n8 n9 n10)

(adds n8 n10 n10)

(adds n9 n0 n9)

(adds n9 n1 n10)

(adds n9 n2 n10)

(adds n9 n3 n10)

(adds n9 n4 n10)

(adds n9 n5 n10)

(adds n9 n6 n10)

(adds n9 n7 n10)

(adds n9 n8 n10)

(adds n9 n9 n10)

(adds n9 n10 n10)

(adds n10 n0 n10)

(adds n10 n1 n10)

(adds n10 n2 n10)

(adds n10 n3 n10)

(adds n10 n4 n10)

(adds n10 n5 n10)

(adds n10 n6 n10)

(adds n10 n7 n10)

(adds n10 n8 n10)

(adds n10 n9 n10)

(adds n10 n10 n10)

(adjacent re ree) (adjacent ree re) ;sosednie komnati

(adjacent rc rn) (adjacent rn rc)

(room\_has\_container rn cn) ; v komnate

(room\_has\_container rs cs)

(room\_has\_container ree cee)

(room\_floor\_coins rc n2)

(room\_floor\_coins rn n2)

(container\_has\_coins cn n2)

(container\_has\_coins cs n5)

(container\_has\_coins cee n1)

(robot\_coins robbie n0)

(robot\_at robbie rc)

(vendor\_is\_at vendor rc) ; prodavec naxoditsa vkomante rc

(vendor\_sells\_key vendor ks n2)

(vendor\_sells\_key vendor ke n4)

(key\_unlocks\_door ks rc rs)

(key\_unlocks\_door ke rc re)

)

(:goal (and

(container\_has\_coins cee n6) ; v konteinere doljno bit 6 monet

))

)

**PROBLEM 3**

И так , у нас есть два робота : Робби и Шаки , два продавца . Один робот начинает свой путь с 5ой комнаты и второй робот начинает с 1 , их цель встретиться в центральной комнате , то есть в 3ой . И соответственно в центральной комнате должно быть 6 монет.

(define (problem coins1)

(:domain coins)

(:objects

ree - room

re - room

rc - room

rw - room

rww - room

cc - container

robbie - robot

shakey - robot

ke - key

kw - key

ve - vendor

vw - vendor

)

(:init

(adds n0 n0 n0)

(adds n0 n1 n1)

(adds n0 n2 n2)

(adds n0 n3 n3)

(adds n0 n4 n4)

(adds n0 n5 n5)

(adds n0 n6 n6)

(adds n0 n7 n7)

(adds n0 n8 n8)

(adds n0 n9 n9)

(adds n0 n10 n10)

(adds n1 n0 n1)

(adds n1 n1 n2)

(adds n1 n2 n3)

(adds n1 n3 n4)

(adds n1 n4 n5)

(adds n1 n5 n6)

(adds n1 n6 n7)

(adds n1 n7 n8)

(adds n1 n8 n9)

(adds n1 n9 n10)

(adds n1 n10 n10)

(adds n2 n0 n2)

(adds n2 n1 n3)

(adds n2 n2 n4)

(adds n2 n3 n5)

(adds n2 n4 n6)

(adds n2 n5 n7)

(adds n2 n6 n8)

(adds n2 n7 n9)

(adds n2 n8 n10)

(adds n2 n9 n10)

(adds n2 n10 n10)

(adds n3 n0 n3)

(adds n3 n1 n4)

(adds n3 n2 n5)

(adds n3 n3 n6)

(adds n3 n4 n7)

(adds n3 n5 n8)

(adds n3 n6 n9)

(adds n3 n7 n10)

(adds n3 n8 n10)

(adds n3 n9 n10)

(adds n3 n10 n10)

(adds n4 n0 n4)

(adds n4 n1 n5)

(adds n4 n2 n6)

(adds n4 n3 n7)

(adds n4 n4 n8)

(adds n4 n5 n9)

(adds n4 n6 n10)

(adds n4 n7 n10)

(adds n4 n8 n10)

(adds n4 n9 n10)

(adds n4 n10 n10)

(adds n5 n0 n5)

(adds n5 n1 n6)

(adds n5 n2 n7)

(adds n5 n3 n8)

(adds n5 n4 n9)

(adds n5 n5 n10)

(adds n5 n6 n10)

(adds n5 n7 n10)

(adds n5 n8 n10)

(adds n5 n9 n10)

(adds n5 n10 n10)

(adds n6 n0 n6)

(adds n6 n1 n7)

(adds n6 n2 n8)

(adds n6 n3 n9)

(adds n6 n4 n10)

(adds n6 n5 n10)

(adds n6 n6 n10)

(adds n6 n7 n10)

(adds n6 n8 n10)

(adds n6 n9 n10)

(adds n6 n10 n10)

(adds n7 n0 n7)

(adds n7 n1 n8)

(adds n7 n2 n9)

(adds n7 n3 n10)

(adds n7 n4 n10)

(adds n7 n5 n10)

(adds n7 n6 n10)

(adds n7 n7 n10)

(adds n7 n8 n10)

(adds n7 n9 n10)

(adds n7 n10 n10)

(adds n8 n0 n8)

(adds n8 n1 n9)

(adds n8 n2 n10)

(adds n8 n3 n10)

(adds n8 n4 n10)

(adds n8 n5 n10)

(adds n8 n6 n10)

(adds n8 n7 n10)

(adds n8 n8 n10)

(adds n8 n9 n10)

(adds n8 n10 n10)

(adds n9 n0 n9)

(adds n9 n1 n10)

(adds n9 n2 n10)

(adds n9 n3 n10)

(adds n9 n4 n10)

(adds n9 n5 n10)

(adds n9 n6 n10)

(adds n9 n7 n10)

(adds n9 n8 n10)

(adds n9 n9 n10)

(adds n9 n10 n10)

(adds n10 n0 n10)

(adds n10 n1 n10)

(adds n10 n2 n10)

(adds n10 n3 n10)

(adds n10 n4 n10)

(adds n10 n5 n10)

(adds n10 n6 n10)

(adds n10 n7 n10)

(adds n10 n8 n10)

(adds n10 n9 n10)

(adds n10 n10 n10)

(adjacent rww rw) (adjacent rw rww)

(adjacent ree re) (adjacent re ree)

(room\_has\_container rc cc)

(room\_floor\_coins rww n5)

(room\_floor\_coins re n5)

(container\_has\_coins cc n0)

(robot\_coins robbie n0)

(robot\_coins shakey n0)

(robot\_at robbie rww)

(robot\_at shakey ree)

(vendor\_is\_at ve re)

(vendor\_sells\_key ve ke n2)

(vendor\_is\_at vw rw)

(vendor\_sells\_key vw kw n2)

(key\_unlocks\_door ke re rc)

(key\_unlocks\_door kw rc rw)

)

(:goal (and

(container\_has\_coins cc n6) ;v centralnoi konteinere doljnobit 6monet

))

)