

به نام خدا

پروژه شماره ۲ – هوش مصنوعی پیشرفته

سید سجاد ائمی

شماره دانشجویی: ۹۶۱۳۶۶۱۱۹

استاد: دکتر احد هراتی

استاد حل تمرین: مهندس ویدا عادل

شرح پروژه:

1- فرمول بندی مسئله

برای تعریف فضای بازی، خانه های مربعی شکل را به عنوان گره های یک گراف در نظر گرفتیم. که هر گره به همسایه های چهارگانه خود، متصل می باشد. این اتصال یا به صورت عمودی و یا به صورت افقی می باشد و جهت دار است. به همین منظور از دو predicate برای تعیین اتصال بین گره ها استفاده کردم.

(isup ?u ?d)

این predicate یک رابطه را به این صورت تعریف می کند که بلوک u بالای بلوک d می باشد. و یک اتصال عمودی جهت دار بین این دو بلوک برقرار می باشد.



(isright ?r ?l)

این predicate یک رابطه را به این صورت تعریف می کند که بلوک r سمت راست بلوک l می باشد. و یک اتصال افقی جهت دار بین این دو بلوک برقرار می باشد.



predicate های دیگر:

حضور شی مورد نظر در مکانی خاص (at ?thing ?place)

تعریف یک شی از نوع نگهبان (sentinel ?s)

تعریف یک شی از نوع دیوار (wall ?x)

تعریف یک شی از نوع جعبه (box ?c)

تعریف یک شی از نوع بلوک (block ?b)

2- اکشن ها:

در این مسئله ۸ اکشن داریم که ۴ تای آن مربوط به حرکت نگهبان به ۴ خانه بالا، پایین، چپ و راست می باشد.

GoUp

GoDown

GoLeft

GoRight

و ۴ تای دیگر مربوط به هل دادن جعبه به سمت بالا، پایین، چپ و راست می باشد.

PushUp

PushDown

PushLeft

PushRight

برای مثال در اکشن **GoUp** پارامتر ها، preconditions و effect را بدین صورت تعریف کردم:
parameters (?thing ?from ?to)

یک شی از یک مبدا به یک مقصد می رود:

شرط های لازم:

(isup ?to ?from) مقصد باید در بالای مبدا قرار داشته باشد.

(at ?thing ?from) شی باید در مبدا باشد.

(block ?from) مبدا یک بلوک است.

(block ?to) مقصد یک بلوک است.

(empty ?to) مقصد خالی باشد.

(sentinel ?thing) شی باید نگهبان باشد. زیرا جعبه و دیوار و... نمیتوانند خودشان حرکت نمایند

(not (= ?from ?to)) مبدا و مقصد متفاوت باشند.

تاثیرات اجرای این اکشن:

(at ?thing ?to) شی در مقصد قرار دارد

(not (at ?thing ?from)) شی در مبدا قرار ندارد

(empty ?from) مبدا خالی می باشد

(not (empty ?to)) مقصد خالی نمی باشد

اکشن های PushDown و PushLeft و PushRight به طور مشابه تعریف شده اند.

و در مورد اکشن های هل دادن جعبه، برای مثال اکشن PushRight را بررسی می نماییم:

parameters (?s ?s_place ?c ?c_place ?new_place)

شی s که در مکان s_place قرار دارد، شی c که در مکان c_place قرار دارد را به سمت راست هل می دهد تا شی c در مکان new_place قرار بگیرد.

شرط های لازم:

(isright ?c_place ?s_place) جعبه سمت راست نگهبان باشد
(isright ?new_place ?c_place) مقصد جعبه در سمت راست جعبه باشد
(at ?c ?c_place) جعبه در مکان اولیه خود باشد
(at ?s ?s_place) نگهبان در مکان اولیه خود باشد
(block ?s_place) مکان نگهبان یک بلوک می باشد
(block ?c_place) مکان جعبه یک بلوک می باشد
(block ?new_place) مقصد جعبه یک بلوک می باشد
(empty ?new_place) مقصد خالی باشد
(sentinel ?s) هل دهنده باید نگهبان باشد
(box ?c) شی که هل داده می شود باید جعبه باشد
(not (= ?c_place ?new_place)) مبدا و مقصد جعبه برابر نباشند
(not (= ?s_place ?c_place)) نگهبان و جعبه در یک خانه نمی باشند
(not (= ?s_place ?new_place)) نگهبان در مقصد جعبه نباشد

تاثیرات اجرای این اکشن:

(at ?s ?c_place) نگهبان به مکان قبلی جعبه می رود
(not (at ?s ?s_place)) نگهبان در مکان قبلی خود نیست
(at ?c ?new_place) جعبه در مکان جدید خود می باشد
(not (at ?c ?c_place)) جعبه در مکان قبلی خود نیست
(empty ?s_place) مکان قبلی نگهبان خالی می باشد
(not (empty ?new_place)) مکان جدید جعبه خالی نیست

اکشن های PushUp و PushDown و PushLeft به طور مشابه تعریف شده اند.

3- تعریف problem:

در این حالت برای تعریف نام بلوک ها از ترتیب زیر استفاده شده است:

aa	ab	ac	ad
ba	bb	bc	bd
ca	cb	cc	cd
da	db	dc	dd

	X	C	S

بدین ترتیب تعریف حالت init مسئله به صورت زیر می شود:

(:init

(wall X) (at X bb)
 (sentinel S) (at S bd)
 (box C) (at C bc)

(isup aa ba)
 (isup ba ca)
 (isup ca da)

(isup ab bb)
 (isup bb cb)
 (isup cb db)

(isup ac bc)
 (isup bc cc)
 (isup cc dc)

(isup ad bd)
 (isup bd cd)
 (isup cd dd)

(isright ab aa)
 (isright bb ba)
 (isright cb ca)
 (isright db da)

(isright ac ab)
 (isright bc bb)
 (isright cc cb)
 (isright dc db)

(isright ad ac)
 (isright bd bc)
 (isright cd cc)
 (isright dd dc)

(block aa) (empty aa)
 (block ab) (empty ab)
 (block ac) (empty ac)
 (block ad) (empty ad)
 (block ba) (empty ba)
 (block bb) (not (empty bb))

(block bc) (not (empty bc))
 (block bd) (not (empty bd))
 (block ca) (empty ca)
 (block cb) (empty cb)
 (block cc) (empty cc)
 (block cd) (empty cd)
 (block da) (empty da)
 (block db) (empty db)
 (block dc) (empty dc)
 (block dd) (empty dd)

)

و حالت goal به این شکل تعریف می شود:

(:goal (and (at C ba)))

حال به سراغ دو نمونه اصلی می رویم:

نمونه اول:

		X	X
		X	X
C	S		
		C	
		X	X

تعریف problem:

```

(define (problem test1)
  (:domain besco)
  (:objects
    aa ab ac ad
    ba bb bc bd
    ca cb cc cd
    da db dc dd
    ea eb ec ed
    X1 X2 X3 X4 X5 X6 C1 C2 S)

  (:goal (and (at C1 ab) (at C2 ca)))
  (:init

    (wall X1) (at X1 ac)
    (wall X2) (at X2 ad)
    (wall X3) (at X3 bc)
    (wall X4) (at X4 bd)
    (wall X5) (at X5 ec)
  )

```

(wall X6) (at X6 ed)

(sentinel S) (at S cb)

(box C1) (at C1 ca)

(box C2) (at C2 dc)

(isup aa ba)

(isup ba ca)

(isup ca da)

(isup da ea)

(isup ab bb)

(isup bb cb)

(isup cb db)

(isup db eb)

(isup ac bc)

(isup bc cc)

(isup cc dc)

(isup dc ec)

(isup ad bd)

(isup bd cd)

(isup cd dd)

(isup dd ed)

(isright ab aa)

(isright bb ba)

(isright cb ca)

(isright db da)

(isright eb ea)

(isright ac ab)

(isright bc bb)

(isright cc cb)

(isright dc db)

(isright ec eb)

(isright ad ac)

(isright bd bc)

(isright cd cc)

(isright dd dc)

(isright ed ec)

(block aa) (empty aa)

(block ab) (empty ab)

(block ac)

(block ad)

(block ba) (empty ba)

(block bb) (empty bb)

(block bc)

(block bd)

(block ca)

(block cb)

(block cc) (empty cc)

(block cd) (empty cd)

(block da) (empty da)

(block db) (empty db)

(block dc)

(block dd) (empty dd)

(block ea) (empty ea)
(block eb) (empty eb)
(block ec)
(block ed)

))

برای این مسئله، خروجی Plan به صورت زیر می باشد:

Found Plan (output)

(godown s cb db)
(goleft s db da)
(pushup s da c2 ca ba)
(goright s ca cb)
(goright s cb cc)
(goright s cc cd)
(godown s cd dd)
(pushleft s dd c1 dc db)
(goright s dc dd)
(goup s dd cd)
(goleft s cd cc)
(goleft s cc cb)
(goleft s cb ca)
(godown s ca da)
(godown s da ea)
(goright s ea eb)
(pushup s eb c1 db cb)
(godown s db eb)
(goleft s eb ea)
(goup s ea da)(goup s da ca)
(pushright s ca c1 cb cc)
(goup s cb bb)
(goup s bb ab)
(goleft s ab aa)
(pushdown s aa c2 ba ca)
(goright s ba bb)
(godown s bb cb)
(godown s cb db)
(goright s db dc)
(goright s dc dd)
(goup s dd cd)
(pushleft s cd c1 cc cb)
(godown s cc dc)
(goleft s dc db)
(pushup s db c1 cb bb)
(pushup s cb c1 bb ab)

S	X			C	
S	X		X	C	
	X				X
	X		X		X
	X		X	C	X
			X		X

تعريف problem:

```
(define (problem test3)
  (:domain besco)
  (:objects aa ab ac ad ae af
            ba bb bc bd be bf
            ca cb cc cd ce cf
            da db dc dd de df
            ea eb ec ed ee ef
            fa fb fc fd fe ff
            X1 X2 X3 X4 X5 X6 X7 X8 X9 X10 X11 X12 X13
            C1 C2 C3
            S1 S2)

  (:goal (and (at C1 af) (at C2 ce) (at C3 fe)))

  (:init

    (wall X1) (at X1 ab)
    (wall X2) (at X2 bb)
    (wall X3) (at X3 cb)
    (wall X4) (at X4 db)
    (wall X5) (at X5 eb)
    (wall X6) (at X6 bd)
    (wall X7) (at X7 dd)
    (wall X8) (at X8 ed)
    (wall X9) (at X9 fd)
    (wall X10) (at X10 cf)
    (wall X11) (at X11 df)
    (wall X12) (at X12 ef)
    (wall X13) (at X13 ff)

    (sentinel S1) (at S1 aa)
    (sentinel S2) (at S2 ba)

    (box C1) (at C1 ae)
    (box C2) (at C2 be)
    (box C3) (at C3 ee)

    (isup aa ba)
    (isup ba ca)
    (isup ca da)
    (isup da ea)
```


(isup ea fa)

(isup ab bb)

(isup bb cb)

(isup cb db)

(isup db eb)

(isup eb fb)

(isup ac bc)

(isup bc cc)

(isup cc dc)

(isup dc ec)

(isup ec fc)

(isup ad bd)

(isup bd cd)

(isup cd dd)

(isup dd ed)

(isup ed fd)

(isup ae be)

(isup be ce)

(isup ce de)

(isup de ee)

(isup ee fe)

(isup af bf)

(isup bf cf)

(isup cf df)

(isup df ef)

(isup ef ff)

(isright ab aa)

(isright bb ba)

(isright cb ca)

(isright db da)

(isright eb ea)

(isright fb fa)

(isright ac ab)

(isright bc bb)

(isright cc cb)

(isright dc db)

(isright ec eb)

(isright fc fb)

(isright ad ac)

(isright bd bc)

(isright cd cc)

(isright dd dc)

(isright ed ec)

(isright fd fc)

(isright ae ad)

(isright be bd)

(isright ce cd)

(isright de dd)

(isright ee ed)

(isright fe fd)

(isright af ae)

(isright bf be)
(isright cf ce)
(isright df de)
(isright ef ee)
(isright ff fe)

(block aa)
(block ab)
(block ac) (empty ac)
(block ad) (empty ad)
(block ae)
(block af) (empty af)
(block ba)
(block bb)
(block bc) (empty bc)
(block bd)
(block be)
(block bf) (empty bf)
(block ca) (empty ca)
(block cb)
(block cc) (empty cc)
(block cd) (empty cd)
(block ce) (empty ce)
(block cf)
(block da) (empty da)
(block db)
(block dc) (empty dc)
(block dd)
(block de) (empty de)
(block df)
(block ea) (empty ea)
(block eb)
(block ec) (empty ec)
(block ed)
(block ee)
(block ef)
(block fa) (empty fa)
(block fb) (empty fb)
(block fc) (empty fc)
(block fd)
(block fe) (empty fe)
(block ff)

))

برای این مسئله، خروجی Plan به صورت زیر می باشد:

Found Plan (output)

(godown s2 ba ca)
(godown s2 ca da)
(godown s2 da ea)
(godown s2 ea fa)
(goright s2 fa fb)
(goright s2 fb fc)
(goup s2 fc ec)
(goup s2 ec dc)
(goup s2 dc cc)
(goright s2 cc cd)
(goright s2 cd ce)

(godown s2 ce de)
(pushdown s2 de c3 ee fe)
(goup s2 ee de)
(goup s2 de ce)
(goleft s2 ce cd)
(goleft s2 cd cc)
(goup s2 cc bc)
(goup s2 bc ac)
(goright s2 ac ad)
(pushright s2 ad c1 ae af)
(pushdown s2 ae c2 be ce)