Planning

Introduction to STRIPS

The Block domain:

- 3 blocks (a, b, c)
- 4 table positions $(1, 2, 3, 4)$
- 1 action: *move(X, From, To)* - grab the block X on position *From* and release it on position *To*. Both positions should be visible from the top; we denote that by relation *clear(Pos)*. The relation *on(X, Pos)* denotes that block X in on position *Pos*.

The initial state is shown in Fig. 1. Formally, we write:

$$state = [clear(2), clear(4), clear(b), clear(c), on(a, 1), on(c, a), on(b, 3)].$$

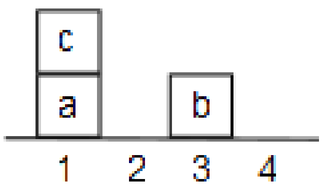

Figure 1: Začetno stanje v svetu kock

Definition of action *move*

A = *move(X, From, To)*

can: in the current state, all preconditions must be satisfied to make the action executable,

can(A) = [clear(X), clear(To), on(X,From)].

adds: (positive effects) the relations that the action establishes in the new state upon its execution,

adds(A) = [clear(From), on(X,To)].

dels: (negative effects) the relations that are no longer true in the new state after the action is performed,

dels(A) = [clear(To), on(X,From)].

constraints: only to prune the search space.

constraints(A) = [$X \neq From, X \neq To, To \neq From$, *block(X)*].

The constraint $X \neq To$ prevents moving X to itself, $To \neq From$ prevents moving to the same position, $block(X)$ demands that X is a block and prevents moving table positions 1,2,3,4.

Goal regression

Pri metodi sredstva-cilji se posamezni cilji rešujejo lokalno, kar mnogokrat onemogoča poiskati najkrajšo rešitev. To imenujemo Sussmanova anomalija. Regresiranje ciljev se problema loti globalno in poskuša doseči vse cilje naenkrat. S tem omogoča iskanje optimalnih planov.

Postopek začnemo tako, da iz množice ciljev izberemo cilj in ustrezno akcijo, s katero bi ta cilj dosegli. V naslednjem koraku se vprašamo, kaj vse bi moralo veljati, da bi bili po izvedeni izbrani akciji doseženi vsi cilji (ne samo izbrani). Odgovor na to vprašanje je v regresiji ciljev, ki iz prejšnjih ciljev, pogojev za akcijo in učinkov akcije izračuna nove cilje. Od tu naprej nas zanimajo le ti novi cilji, saj če jih uresničimo, bomo na koncu z izbrano akcijo rešili vse začetne cilje. Postopek se rekurzivno ponavlja dokler ne dobimo množice ciljev, ki so izpolnjeni že v začetni poziciji.

Algoritem (na nivoju i; na začetku so cilji $goals(0)$ enaki končnim ciljem):

1. Če so vsi cilji v $goals(i)$ resnični v začetnem stanju, končamo in vse akcije izvedemo v obratnem vrstnem redu. Če $goals(i)$ ni možno doseči (nemogoči cilji ali ni primerne akcije), se vrnemo v prostoru stanj in poskušamo najti rešitev po drugi poti.

2. Če cilji $goals(i)$ še niso uresničeni in so izvedljivi, izberemo cilj iz $goals(i)$ in akcijo A, ki doda ta cilj in regresiramo cilje po naslednji formuli:

$$goals(i+1) = goals(i) \cup conditions(A) \setminus adds(A)$$

Pri tem moramo paziti, da A ne izbriše trenutnega cilja (v $del(A)$ ni cilja iz $goals(i)$, oz. presek $del(A)$ in $goals(i)$ je prazna množica).

Naloga g: simuliranje algoritma z regresiranjem ciljev

Za zagotavljanje najkrajše rešitve uporabimo iterativno poglabljanje. Začnemo pri $D = 3$, globini $D = 1$ in $D = 2$ zaradi preglednosti izpustimo.

$goals(0) = [on(a, b), on(b, c)]$

Ali so cilji $on(a, b), on(b, c)$ izpolnjeni v začetni poziciji? Ne. Izberemo cilj: $on(a, b)$. Akcija: $move(a, From, b)$, predpogoji so:

$conditions(move(a, From, b)) = [clear(a), clear(b), on(a, From)]$.

Možne vrednosti za $From$ so: $1, 2, 3, 4, c$. Nastavimo $From = 1$.

$adds(move(a, 1, b)) = [clear(1), on(a, b)]$
$dels(move(a, 1, b)) = [clear(b), on(a, 1)]$

Akcijo lahko izvedemo, ker $dels$ ne vsebuje cilja iz $goals(0)$.

Regresija ciljev:

$$goals(1) = goals(0) \cup conditions(move(a, 1, b)) \setminus adds(move(a, 1, b)) =$$
$$= [on(a, b), on(b, c)] \cup [clear(a), clear(b), on(a, 1)] \setminus [clear(1), on(a, b)] =$$
$$= [clear(a), clear(b), on(a, 1), on(b, c)].$$

Regresirane cilje interpretiramo takole: če lahko pridemo v stanje, kjer velja $goals(1)$, bomo z akcijo $move(a, 1, b)$ prišli v stanje, kjer velja $goals(0)$. Postopek nadaljujemo, dokler $goals(i)$ niso izpolnjeni v začetnem stanju.

$$goals(1) = [clear(a), clear(b), on(a, 1), on(b, c)]$$

Ali so novi cilji $goals(1)$ izpolnjeni v začetni poziciji? Ne. Izberemo cilj: $clear(a)$ iz $goals(1)$ (po vrsti) in izberemo akcijo $move(X, a, To)$. Predpogoji so:

$$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)],$$

možne vrednosti za X in To so: $(c, 2)$,$(c, 4)$, (c, b), $(c, 1)$, itd. Izberemo $X = c$, $To = 2$.

$$adds(move(c, a, 2)) = [clear(a), on(c, 2)]$$
$$dels(move(c, a, 2)) = [clear(2), on(c, a)]$$

Akcijo lahko izvedemo, ker $clear(2)$ in $on(c, a)$ nista v trenutni množici ciljev $goals(1)$.

Regresija ciljev:

$$goals(2) = goals(1) \cup conditions(move(c, a, 2)) \setminus adds(move(c, a, 2)) =$$
$$= [clear(a), clear(b), on(a, 1), on(b, c)] \cup [clear(c), clear(2), on(c, a)] \setminus$$
$$[clear(a), on(c, 2)] =$$
$$= [clear(c), clear(2), on(c, a), clear(b), on(a, 1), on(b, c)].$$

Tu ne moremo nadaljevati, saj cilj ni izvedljiv! Hkrati ni možno doseči ciljev $on(b, c)$ in $clear(c)$.

Zdaj lahko poskusimo z drugimi vrednostmi spremenljivk X in To, vendar ne bi našli rešitve v treh ali manj korakih. Vrnemo se korak nazaj; izbrati moramo nov cilj.

Cilja $clear(b)$ in $on(a, 1)$ v $goals(1)$ sta v začetnem stanju že resnična, izberemo cilj: $on(b, c)$. Akcija $move(b, From, c)$, predpogoji:

$$conditions(move(b, From, c)) = [clear(b), clear(c), on(b, From)].$$

Možne vrednosti za $From$ so: $3, 1, 2, 4, a$. Nastavimo $From = 3$.

$$adds(move(b, 3, c)) = [clear(3), on(b, c)]$$
$$dels(move(b, 3, c)) = [clear(c), on(b, 3)]$$

V $dels$ ni relacije, ki bi bila tudi v trenutnih ciljih, torej lahko izvedemo akcijo.

Regresija ciljev:

$goals(2) = goals(1) \cup conditions(move(b, 3, c)) \setminus adds(move(b, 3, c)) =$
$= [clear(a), clear(b), on(a, 1), on(b, c)] \cup [clear(b), clear(c), on(b, 3)] \setminus$
$[clear(3), on(b, c)] = = [clear(a), clear(b), clear(c), on(a, 1), on(b, 3)]$.

$$goals(2) = [clear(a), clear(b), clear(c), on(a, 1), on(b, 3)]$$

Ali so cilji $goals(2)$ resnični v začetnem stanju? Ne. Izberemo cilj: $clear(a)$, izberemo akcijo $move(X, a, To)$, njeni predpogoji so:

$conditions(move(X, a, To)) = [clear(X), clear(To), on(X, a)]$,

vrednosti za X in To so: $(c, 2), (c, 4), (c, b), ...$ Nastavimo vrednosti $X = c$, $To = 2$.

$adds(move(c, a, 2)) = [clear(a), on(c, 2)]$
$dels(move(c, a, 2)) = [clear(2), on(c, a)]$

Med cilji v $goals(2)$ in $dels(move(c, a, 2))$ ni konflikta .

Regresija ciljev:

$goals(3) = goals(2) \cup conditions(move(c, a, 2)) \setminus adds(move(c, a, 2)) =$
$= [clear(a), clear(b), clear(c), on(a, 1), on(b, 3)] \cup$
$[clear(c), clear(2), on(c, a)] \setminus [clear(a), on(c, 2)] =$
$= [clear(b), clear(c), on(a, 1), on(b, 3), clear(2), on(c, a)]$.

$$goals(3) = [clear(b), clear(c), on(a, 1), on(b, 3), clear(2), on(c, a)]$$

Opazimo, da so ti cilji resnični v začetnem stanju. Zdaj le še izvedemo akcije v obratnem vrstnem redu. Rešitev je:

1. $move(c, a, 2)$
2. $move(b, 3, c)$
3. $move(a, 1, b)$