Goal (Theorem):

$R(s)$ ...

from location   location
$S: climbs (carry (C, B, walk (A, C, s)))$

to banana   walks to
B   location   next
C   state

next state
for climb

writing the prologue: (These are facts (constraints))

chair at c
$P(a, b, c, s_0)$

monkey at   banana
position a   at position b

initial state

$P(z_1, y_1, z_1, s_2) :- p(x_1, y_1, z_1, s_1), walk(x_1, z_1, s_1, s_2).$ next state

position

monkey same position as chair after it walkes trip
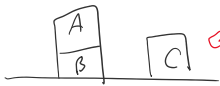
$P(y_2, y_2, y_2, s_4) :- p(x_2, y_2, x_2, s_2), carry(x_2, y_2, s_3, s_4)$

chair and monkey position is same after it came

$r(s_6) :- p(b, b, b, s_5), climbs(s_5, s_6)$
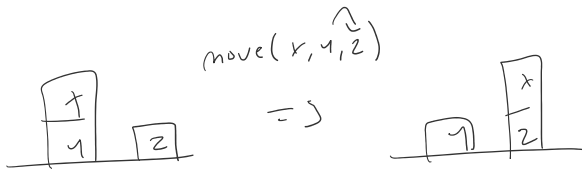
reachable

$? - r(s_6)$

Blocks world

s_0

$On(A, B) \Rightarrow$ A on top of B

$clear(A)$ so

$clear(C)$ so is

This approach is situational calculus

Situational calculus

Frame Problem

$move(x, y, z)$

$\Rightarrow$

positive effect: predicate turned on

y is clear on top now

in this ex. $On(x, z)$, $clear(y)$

$on(x, y, s) \land clear(x, s) \land clear(z, s) \land (x \neq z)$
$\supset On(x, z, do(move(x, y, z), s)))$

positive effect
axiom

negative effect: predicate turned off

in this $on(x, y)$, $clear(z)$

negative true

in this on(v,y) , clear(z)
                ↑   ↑
        this is destroyed   z is no longer
        (no longer)   clear on top
        so it is a negative
        effect   move(x,y,z)

‖
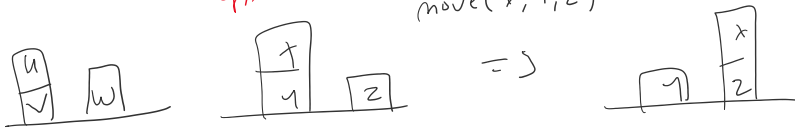
frame axiom   } This is basically a
is has to   frame problem → the
be explicitly   computer can't handle any
stated that   holes, so we must state
u≠v, v≠y,   everything explicitly
etc.

Problem Solver

recursively do it the preconditions
until the current operator the initial
  of at the state

initial state   final state
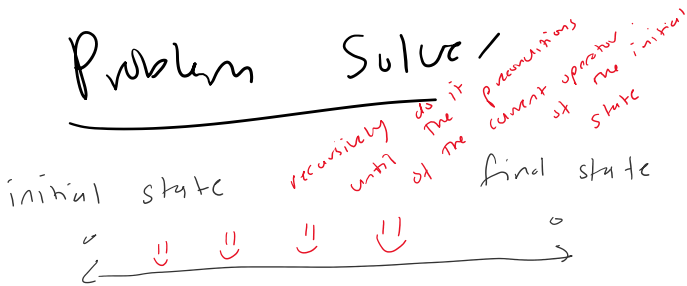
⌐ find an operation whose post conditions can reduce the difference
 └ - the preconditions of the operation may become the new difference

until no difference   - we want the preconditions
                         of the last operator to
                         be satisfied by the
                         initial condition
                       - This is GPS (Global
                                      problem
                                      solver)

Example

① pickup (b).
          P {D : ontable(b), clear(x), hand empty   [b]
                                                    preconditions, have to
                                                    be true
  ↑   ↑
precondition deleted

after it has

after it has
been applied, it
will be deleted

Add list ←— A : holding (b)

② putdown (x):

P & D : Holding (x)
A : Ontable (b), clear (b), hand empty

③ stack (b, y)

P & D : Holding (b), clear (y)
A : Handempty, on (x, y), clear (b)

④ unstack (x, y)

P & D : Hand empty
A : Holding (b), clear (y)

get b from stack and
put it into hand

Initially (I):   Goal: