Q1). a) $f(Marcus)$ and $f(Caesar)$.

Marcus $\neq$ Caesar, constants $\Rightarrow$ fail.

b). $f(x)$ and $f(g(y))$

$x/g(y)$.

c) $f(Marcus, g(x, y))$ and $f(x, g(Caesar, Marcus))$

The 'x' in the two expressions must be treated as separate variable during unification, unless we know otherwise.

$\Rightarrow$ $f(Marcus, g(x_1, y))$ and $f(x_2, g(Caesar, Marcus))$

$\Rightarrow$ $(x_1/Caesar,\ x_2/Marcus,\ y/Marcus)$

Q2

Actions.

clear(y) holding(x)

$\boxed{\text{stack}(x,y)}$

armempty, on(x,y) ¬clear(y) ¬holding(x)

on(x,y) clear(x) armempty

$\boxed{\text{unstack}(x,y)}$

holding(x) clear(y) ¬on(x,y) ¬armempty

clear(x) ontable(x) armempty

$\boxed{\text{pickup}(x)}$

holding(x) ¬ontable(x) ¬armempty

holding(x)

$\boxed{\text{putdown}(x)}$

¬holding(x) ontable(x) armempty

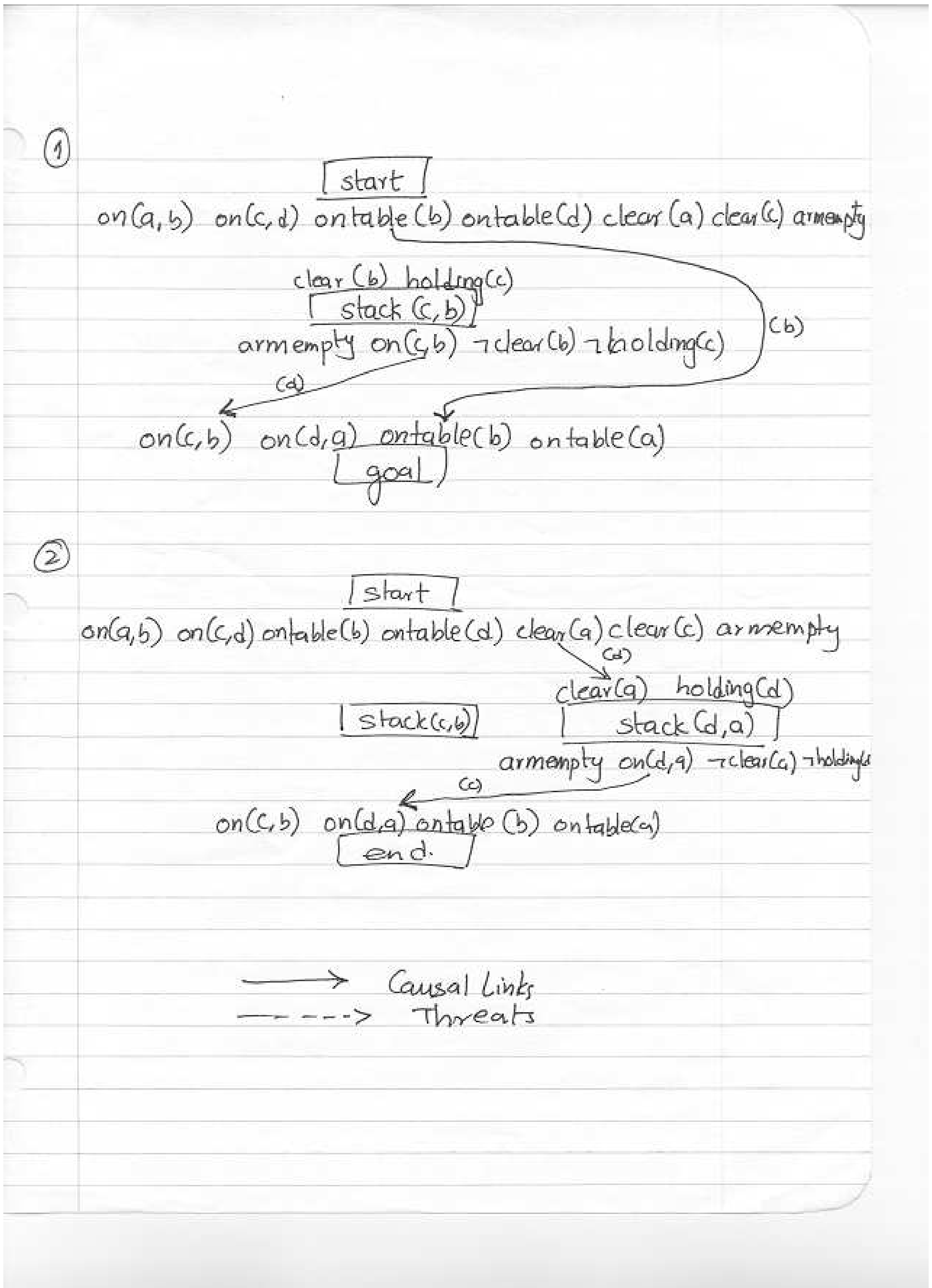
$\boxed{\text{start}}$

on(c,b) on(d,a) ontable(b) ontable(a)
clear(a) clear(c) armempty

on(c,b) on(d,a) ontable(b) ontable(a)

$\boxed{\text{end}}$

POP is a regression planner. So actions that satisfy atleast one un-satisfied pre-condition can be added. Threats are subsequently resolved by promotion or demotion


1
start
on(a,b) on(c,d) ontable(b) ontable(d) clear(a) clear(c) armempty
clear(b) holding(c)
stack(c,b)
armempty on(c,b) ¬clear(b) ¬holding(c)
(b)
(a)
on(c,b) on(d,a) ontable(b) ontable(a)
goal
2
start
on(a,b) on(c,d) ontable(b) ontable(d) clear(a) clear(c) armempty
(d)
clear(a) holding(d)
stack(c,b)
stack(d,a)
armempty on(d,a) ¬clear(a) ¬holding(d)
(c)
on(c,b) on(d,a) ontable(b) ontable(a)
end.
Causal Links
Threats

(3)

start

holding(a)

putdown(a) | stack(c,b) | stack(d,a)

¬holding(a) ontable(a) armempty

(e)

ontable(a) on(c,b) on(d,a) ontable(b)

end.

(4)

start.

on(a,b) on(c,d) ontable(b) ontable(d) clear(a) clear(c) armempty

(g)

clear(d) ontable(d) armempty

pickup(d)

¬ontable(d) holding(d) armempty (f)

clear(a) holding(d)

putdown(a) | stack(c,b) | stack(d,a)

end.

(5)

start

clear(a) armempty on(a,b) ontable(d) ontable(b)

clear(a) on(a,b) armempty

unstack(a,b)

clear(b) ¬on(a,b) holding(a,) ¬armempty

holding(a)

putdown(a)

¬holding(a) armempty ontable(a)

clear(b) holding(c)

stack(c,b)

on(c,b) ¬clear(b) ¬holding(c) arm empty

clear(d) ontable(d) armempty

pickup(d)

¬ontable(d) holding(d) ¬armempty

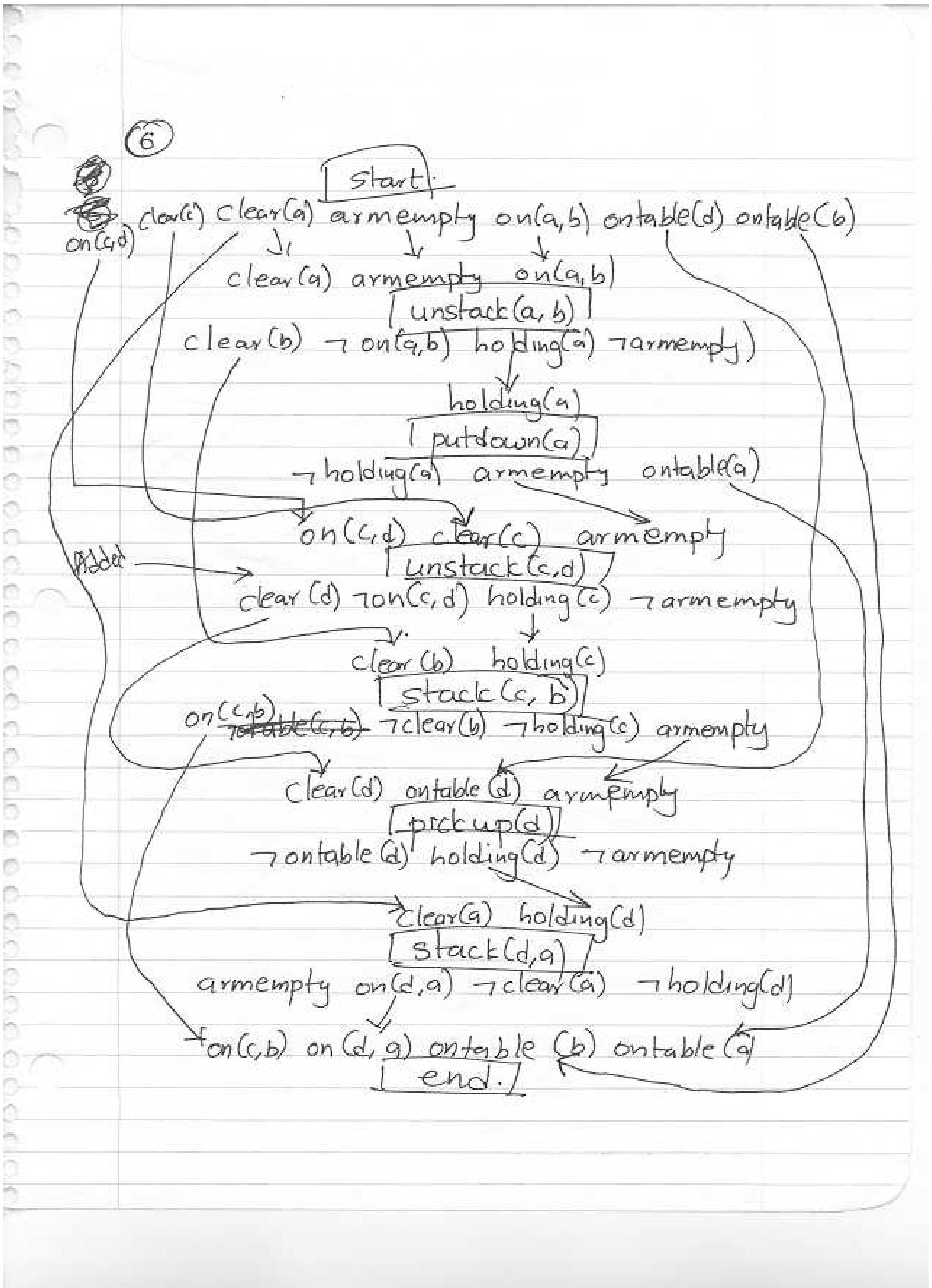
clear(a) holding(d)

stack(d,a)

armempty on(d,a) ¬clear(a) ¬holding(d)

on(c,b) on(d,a) ontable(b) ontable(a)

end.


6
Start
on(c,d) clear(c) clear(a) armempty on(a,b) ontable(d) ontable(b)
clear(a) armempty on(a,b)
unstack(a,b)
clear(b) ¬on(a,b) holding(a) ¬armempty
holding(a)
putdown(a)
¬holding(a) armempty ontable(a)
on(c,d) clear(c) armempty
unstack(c,d)
Added
clear(d) ¬on(c,d) holding(c) ¬armempty
clear(b) holding(c)
stack(c,b)
on(c,b) ~~¬ontable(c,b)~~ ¬clear(b) ¬holding(c) armempty
clear(d) ontable(d) armempty
pickup(d)
¬ontable(d) holding(d) ¬armempty
clear(a) holding(d)
stack(d,a)
armempty on(d,a) ¬clear(a) ¬holding(d)
on(c,b) on(d,a) ontable(b) ontable(a)
end.

Q3

contains(lv, v)   contains(x, ov)

[Assign. (x, v, lv, ov)]

Contains (x, v)   ¬ contains (x, ov)

start: contains (A, a) , contains (B, b) , contains (C, c)

Registers are A, B and C
Values in A and B should be exchanged

Goal: contains (A, b) contains (B, a)

(1)

[start.]

contains (A, a) contains (B, b) contains (C, c)

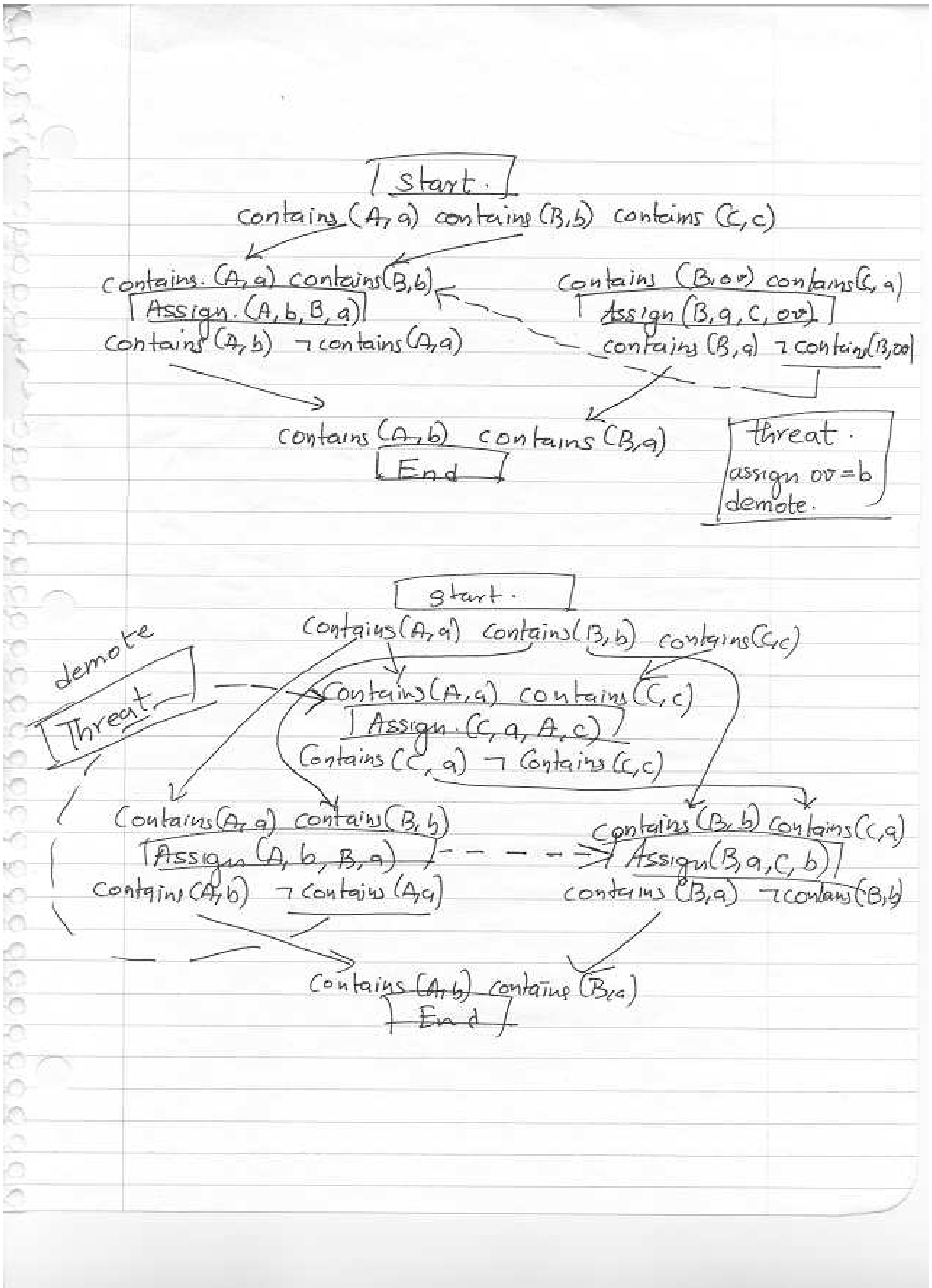
contains (B, ov) contains (C, a)

[Assign (B, a, C, ov)]

Contains (B, a) , ¬ contains (B, ov)

contains (A, b) contains (B, a)

[End]


Start.
contains (A, a) contains (B, b) contains (C, c)
contains. (A, a) contains (B, b)
Assign. (A, b, B, a)
contains (A, b) ¬ contains (A, a)
contains (B, ov) contains (C, a)
Assign (B, a, C, ov)
contains (B, a) ¬ contains (B, ov)
contains (A, b) contains (B, a)
End
threat.
assign ov = b
demote.
start.
contains (A, a) contains (B, b) contains (C, c)
demote
Threat
contains (A, a) contains (C, c)
Assign. (C, a, A, c)
contains (C, a) ¬ contains (C, c)
contains (A, a) contains (B, b)
Assign (A, b, B, a)
contains (A, b) ¬ contains (A, a)
contains (B, b) contains (C, a)
Assign (B, a, C, b)
contains (B, a) ¬ contains (B, b)
contains (A, b) contains (B, a)
End