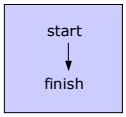
6.825 Techniques in Artificial Intelligence

Partial-Order Planning Algorithms

Partially Ordered Plan

- Plan
  - Steps
  - Ordering constraints
  - Variable binding constraints
  - Causal links
- POP Algorithm
  - Make initial plan



  - Loop until plan is a complete
    - Select a subgoal
    - Choose an operator
    - Resolve threats

Choose Operator

- Choose operator(c, S_{needs})
  - Choose a step S from the plan or a new step S by instantiating an operator that has c as an effect
  - If there's no such step, Fail
  - Add causal link S $\square_c$ S_{needs}
  - Add ordering constraint S < S_{needs}
  - Add variable binding constraints if necessary
  - Add S to steps if necessary

Nondeterministic choice

- Choose – pick one of the options arbitrarily
- Fail – go back to most recent non-deterministic choice and try a different one that has not been tried before

Resolve Threats

- A step S threatens a causal link S_i $\square_c$ S_j iff $\neg$ c $\in$ effects(S) and it's possible that S_i < S < S_j
- For each threat
  - Choose
    - Promote S : S < S_i < S_j
    - Demote S : S_i < S_j < S
  - If resulting plan is inconsistent, then Fail

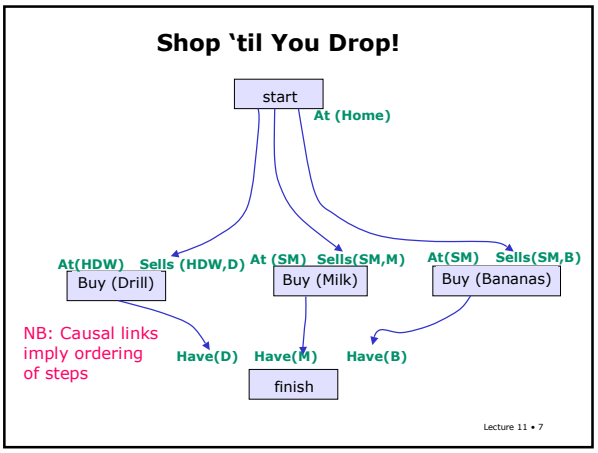
Threats with Variables

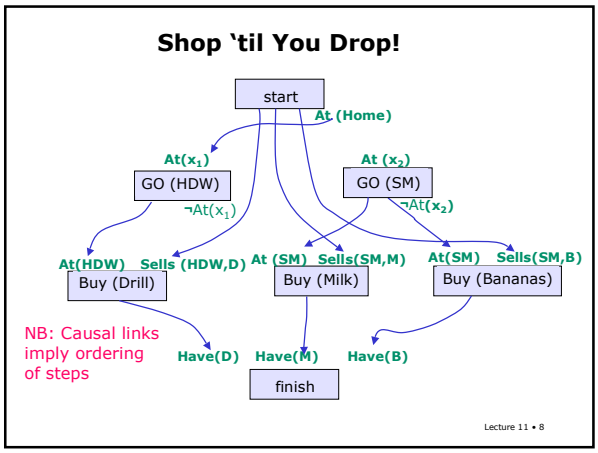
If c has variables in it, things are kind of tricky.

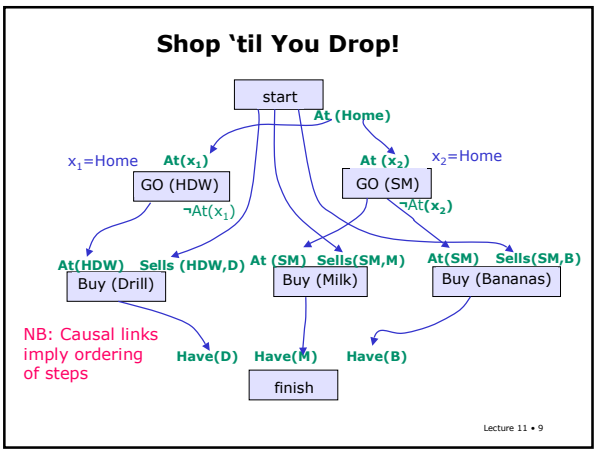
- S is a threat if there is any instantiation of the variables that makes $\neg$ c $\in$ effects(S)
- We could possibly resolve the threat by adding a negative variable binding constraint, saying that two variables or a variable and a constant cannot be bound to one another
- Another strategy is to ignore such threats until the very end, hoping that the variables will become bound and make things easier to deal with

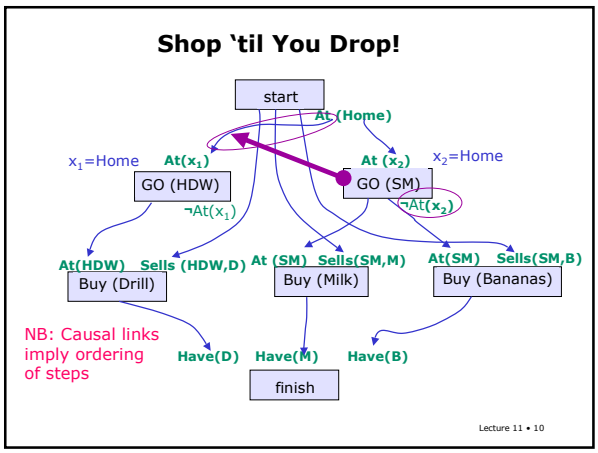
Shopping Domain

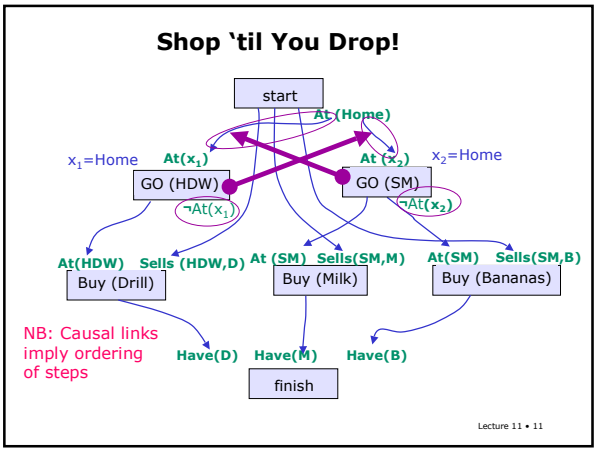
- Actions
  - Buy(x, store)
    - Pre: At(store), Sells(store, x)
    - Eff: Have(x)
  - Go(x, y)
    - Pre: At(x)
    - Eff: At(y), $\neg$At(x)
- Goal
  - Have(Milk) $\wedge$ Have(Banana) $\wedge$ Have(Drill)
- Start
  - At(Home) $\wedge$ Sells(SM, Milk) $\wedge$ Sells(SM, Banana) $\wedge$ Sells(HW, Drill)

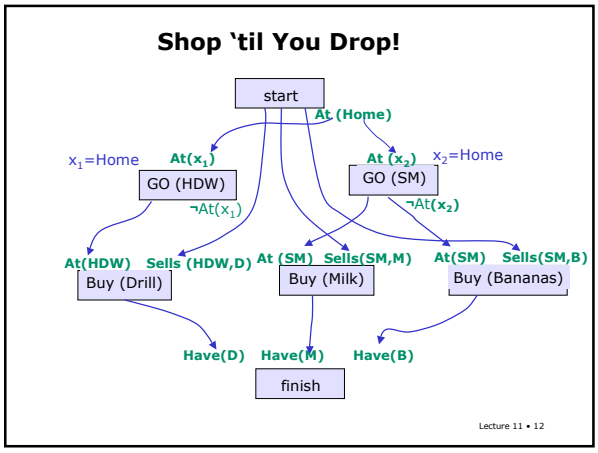

Shop 'til You Drop!
start
At (Home)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
NB: Causal links imply ordering of steps
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 7


Shop 'til You Drop!
start
At (Home)
At(x1)
GO (HDW)
¬At(x1)
At (x2)
GO (SM)
¬At(x2)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
NB: Causal links imply ordering of steps
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 8

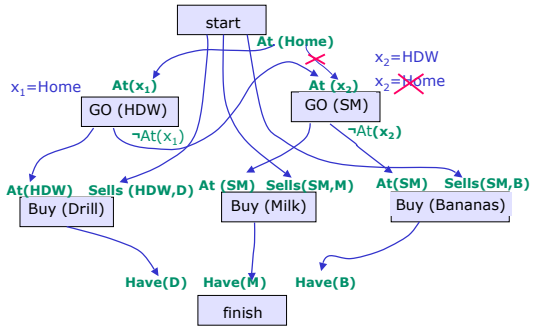

Shop 'til You Drop!
start
At (Home)
x1=Home
At(x1)
GO (HDW)
¬At(x1)
At (x2)
x2=Home
GO (SM)
¬At(x2)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
NB: Causal links imply ordering of steps
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 9


Shop 'til You Drop!
start
At (Home)
x1=Home
At(x1)
GO (HDW)
¬At(x1)
At (x2)
x2=Home
GO (SM)
¬At(x2)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
NB: Causal links imply ordering of steps
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 10

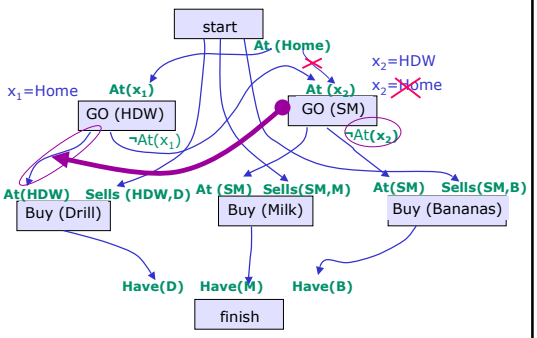

Shop 'til You Drop!
start
At (Home)
x1=Home
At(x1)
GO (HDW)
¬At(x1)
At (x2)
x2=Home
GO (SM)
¬At(x2)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
NB: Causal links imply ordering of steps
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 11


Shop 'til You Drop!
start
At (Home)
x1=Home
At(x1)
GO (HDW)
¬At(x1)
At (x2)
x2=Home
GO (SM)
¬At(x2)
At(HDW) Sells (HDW,D)
Buy (Drill)
At (SM) Sells(SM,M)
Buy (Milk)
At(SM) Sells(SM,B)
Buy (Bananas)
Have(D)
Have(M)
Have(B)
finish
Lecture 11 • 12

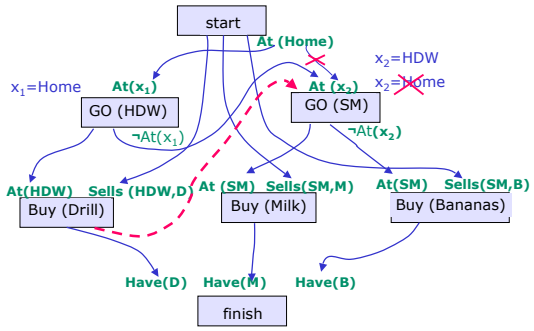
Shop 'til You Drop!



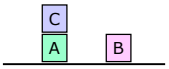
Shop 'til You Drop!



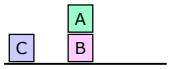
Shop 'til You Drop!



Subgoal Dependence

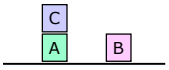


Goal: on(A,B) ∧ on(B,C)

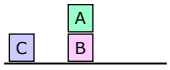


Put A on B first

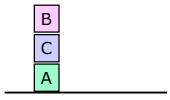
Subgoal Dependence



Objects: A, B, C, Table
on(x,y)
clear(x)
Goal: on(A,B) ∧ on(B,C)



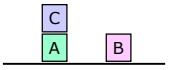
Put A on B first



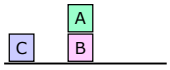
Put B on C first

Subgoal Dependence

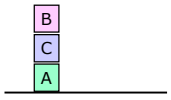
- Move(b, x, y)
  - Pre: on(b,x), clear(b), clear(y), block(y)
  - Eff: on(b,y), clear(x), ¬on(b,x), ¬clear(y)
- Move_table(b, x)
  - Pre: on(b,x), clear(b)
  - Eff: on(b,table), clear(x), ¬on(b,x)



Objects: A, B, C, Table
on(x,y)
clear(x)
Goal: on(A,B) ∧ on(B,C)



Put A on B first



Put B on C first

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

on(A,B) on(B,C)

finish

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

Move (A,x_1,B)

on(A,B) on(B,C)

finish

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

Move (A,x_1,B)

on(A,B) ¬on(A,x_1) clear (x_1) ¬clear(B)

on(A,B) on(B,C)

finish

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

on(A,x_1) clear(A) clear(B)

Move (A,x_1,B)

on(A,B) ¬on(A,x_1) clear (x_1) ¬clear(B)

on(A,B) on(B,C)

finish

Sussman Anomaly

start

clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)

MoveT(y,A)

on(A,x_1) clear(A) clear(B)

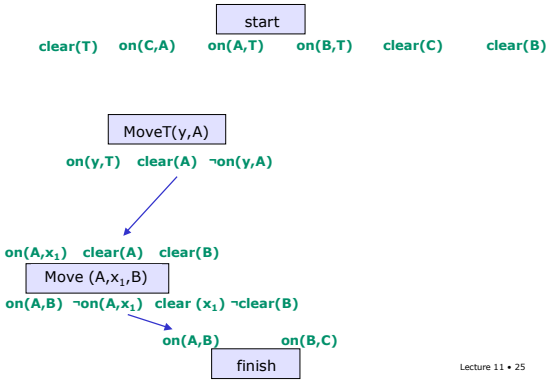
Move (A,x_1,B)

on(A,B) ¬on(A,x_1) clear (x_1) ¬clear(B)

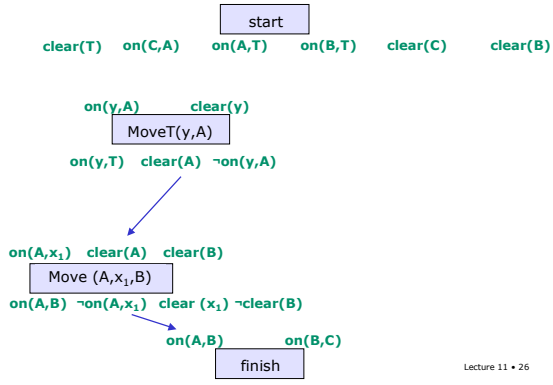
on(A,B) on(B,C)

finish

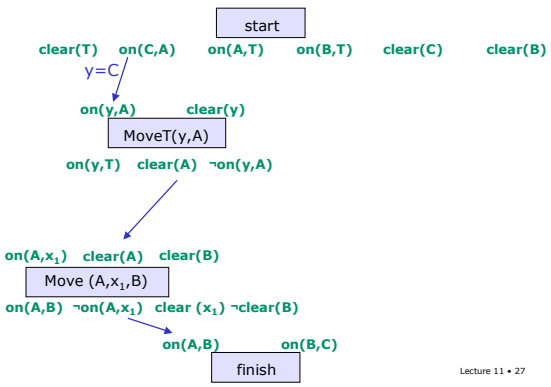
Sussman Anomaly



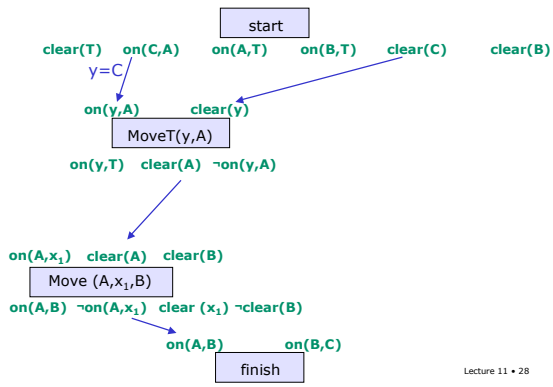
Sussman Anomaly



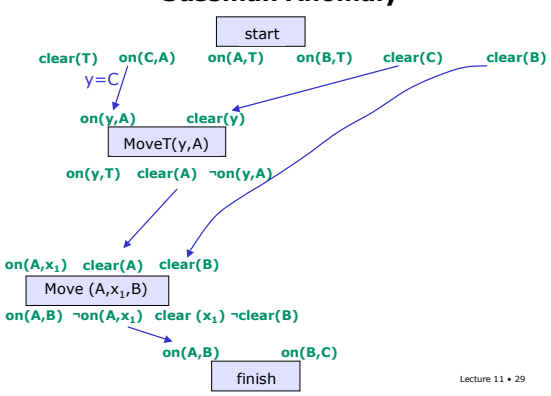
Sussman Anomaly



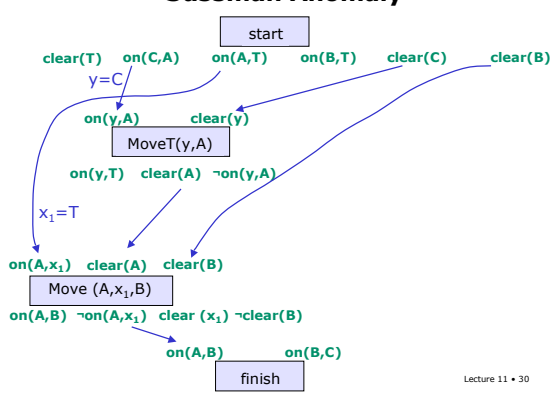
Sussman Anomaly

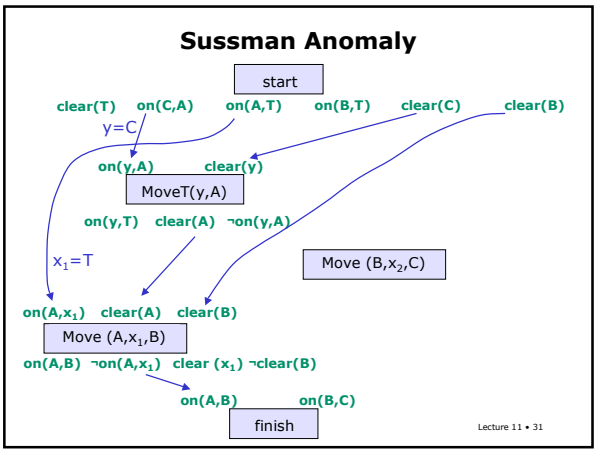


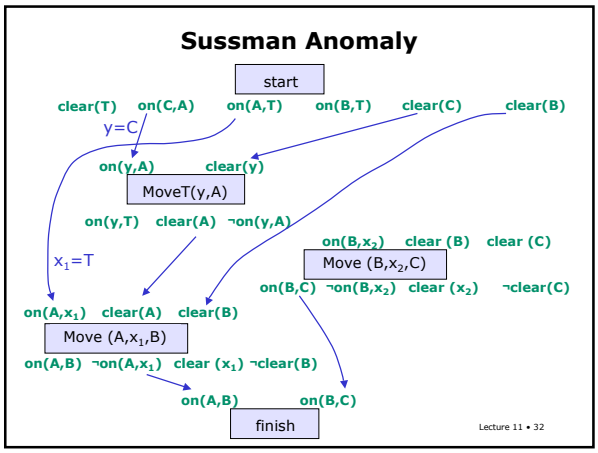
Sussman Anomaly

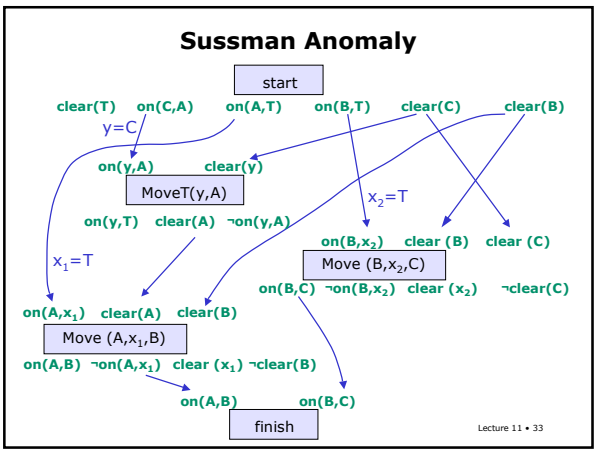


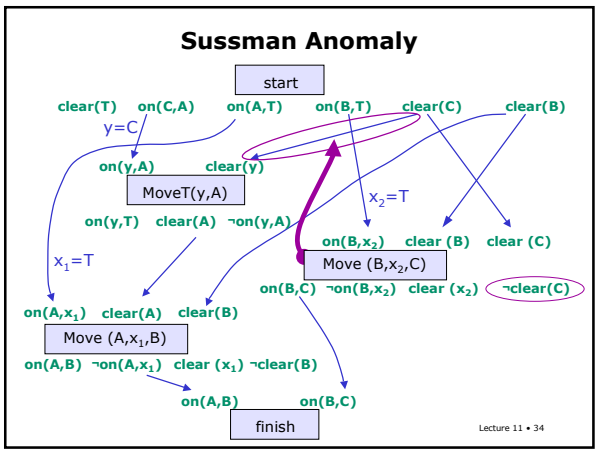
Sussman Anomaly

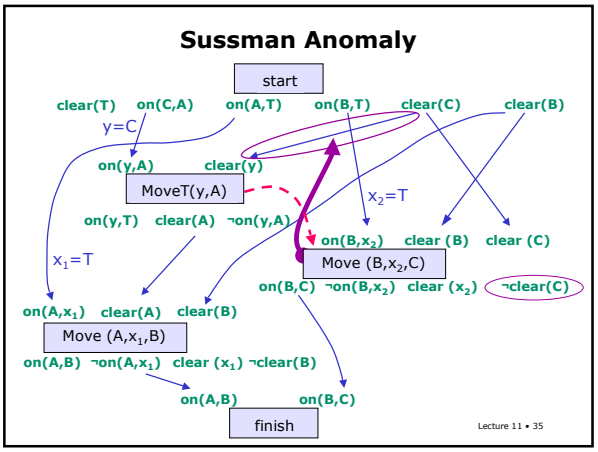


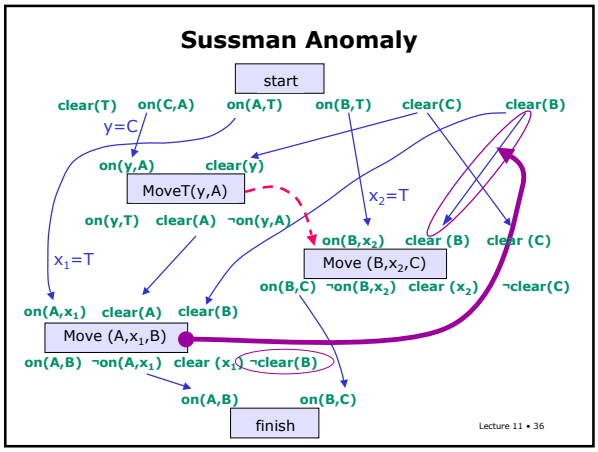

Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
on(y,T) clear(A) ¬on(y,A)
x1=T
Move (B,x2,C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 31


Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
on(y,T) clear(A) ¬on(y,A)
x1=T
on(B,x2) clear (B) clear (C)
Move (B,x2,C)
on(B,C) ¬on(B,x2) clear (x2) ¬clear(C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 32

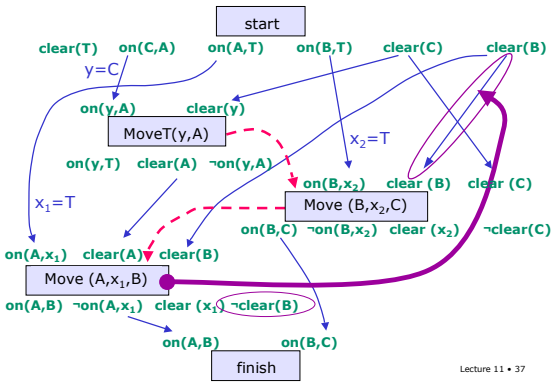

Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
x2=T
on(y,T) clear(A) ¬on(y,A)
x1=T
on(B,x2) clear (B) clear (C)
Move (B,x2,C)
on(B,C) ¬on(B,x2) clear (x2) ¬clear(C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 33


Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
x2=T
on(y,T) clear(A) ¬on(y,A)
x1=T
on(B,x2) clear (B) clear (C)
Move (B,x2,C)
on(B,C) ¬on(B,x2) clear (x2) ¬clear(C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 34

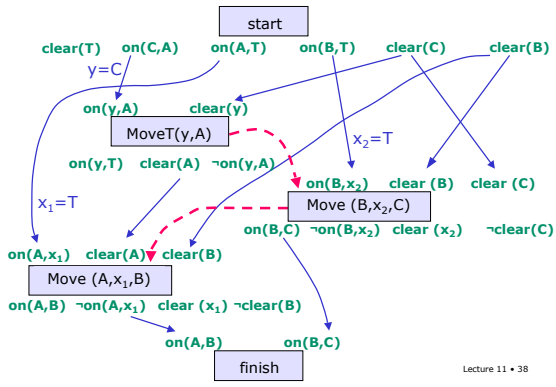

Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
x2=T
on(y,T) clear(A) ¬on(y,A)
x1=T
on(B,x2) clear (B) clear (C)
Move (B,x2,C)
on(B,C) ¬on(B,x2) clear (x2) ¬clear(C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 35


Sussman Anomaly
start
clear(T) on(C,A) on(A,T) on(B,T) clear(C) clear(B)
y=C
on(y,A) clear(y)
MoveT(y,A)
x2=T
on(y,T) clear(A) ¬on(y,A)
x1=T
on(B,x2) clear (B) clear (C)
Move (B,x2,C)
on(B,C) ¬on(B,x2) clear (x2) ¬clear(C)
on(A,x1) clear(A) clear(B)
Move (A,x1,B)
on(A,B) ¬on(A,x1) clear (x1) ¬clear(B)
on(A,B) on(B,C)
finish
Lecture 11 • 36

Sussman Anomaly



Sussman Anomaly



Recitation Problems

- Russell & Norvig
  - 11.2
  - 11.7 a,b,c
  - 11.9