

An early Planner: STRIPS

Representation of operators: A PREcondition list, an ADD list, a DELETE list

pickup(?x): PRE: ONTABLE(?x), CLEAR(?x), HANDEEMPTY
DEL: ONTABLE(?x), CLEAR(?x), HANDEEMPTY
ADD: HOLDING(?x)

putdown(?x): PRE: HOLDING(?x)
DEL: HOLDING(?x)
ADD: ONTABLE(?x), CLEAR(?x), HANDEEMPTY

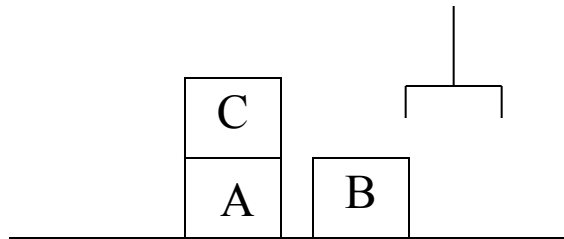
stack(?x, ?y): PRE: HOLDING(?x), CLEAR(?y)
DEL: HOLDING(?x), CLEAR(?y)
ADD: HANDEEMPTY, ON(?x, ?y), CLEAR(?x)

unstack(?x, ?y): PRE: HANDEEMPTY, CLEAR(?x), ON(?x,?y)
DEL: HANDEEMPTY, CLEAR(?x), ON(?x,?y)
ADD: HOLDING(?x), CLEAR(?y)

$OP(S[t1]) = S[t1] - DEL(OP) + ADD(OP) = S[t2]$, where PRE(OP) is a subset of S[t1]

State Description

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY



OP

unstack(?x,?y)
PRE: HANDEEMPTY,
CLEAR(?x),
ON(?x, ?y)
DEL: HANDEEMPTY,
CLEAR(?x),
ON(?x, ?y)
ADD: HOLDING(?x),
CLEAR(?y)

Resulting State

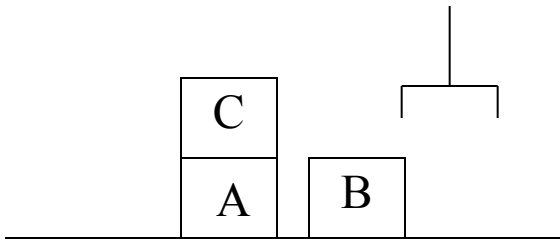
State Description

OP

Resulting State

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

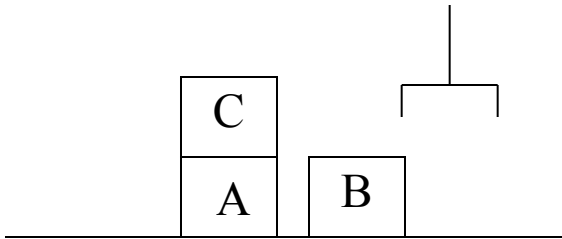
unstack(?x,?y)
PRE: HANDEEMPTY,
CLEAR(?x), {?x/C}
ON(?x, ?y) {?y/A}
DEL: HANDEEMPTY,
CLEAR(?x),
ON(?x, ?y)
ADD: HOLDING(?x),
CLEAR(?y)



Unify (ONTABLE(?x,?y), ONTABLE(C,A))

State Description

CLEAR(B)
CLEAR(C)
ON(C,A)
ONTABLE(A)
ONTABLE(B)
HANDEEMPTY

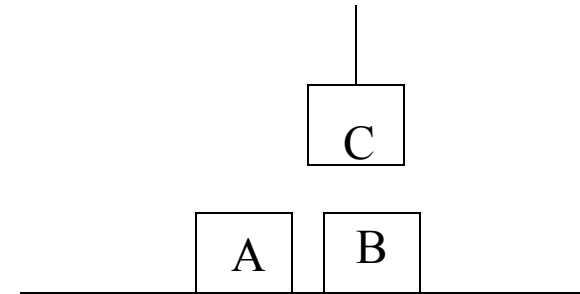


OP

unstack(C,A)
PRE: HANDEEMPTY,
CLEAR(C),
ON(C,A)
DEL: HANDEEMPTY,
CLEAR(C),
ON(C,A)
ADD: HOLDING(C),
CLEAR(A)

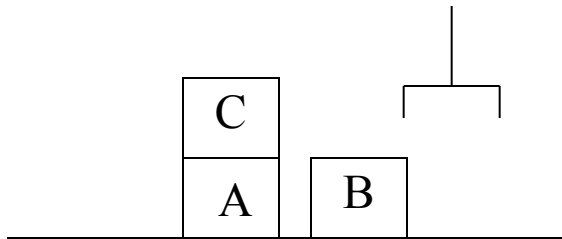
Resulting State

CLEAR(B)
~~CLEAR(C)~~
~~ON(C,A)~~
ONTABLE(A)
ONTABLE(B)
~~HANDEEMPTY~~
HOLDING(C)
CLEAR(A)



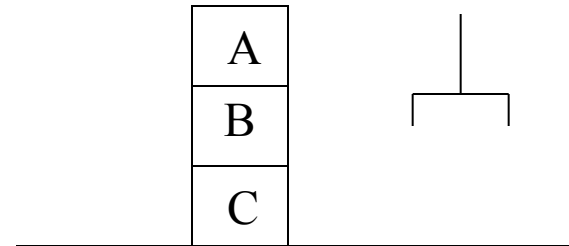
Planning problem: Given an **initial state** and **goal specification**,
find a sequence/set of operators that transform initial state to a state that
satisfies goal specification

Initial State

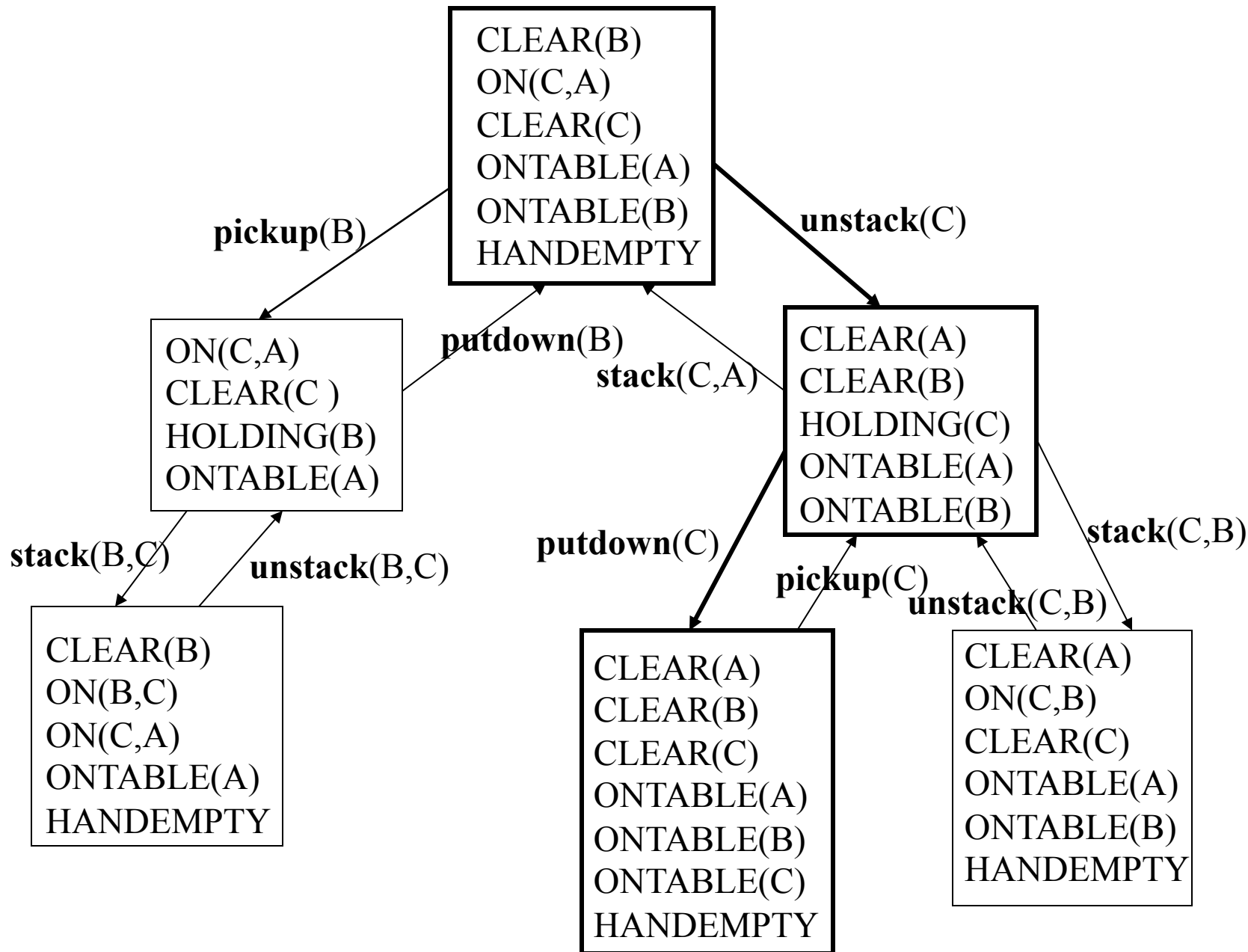


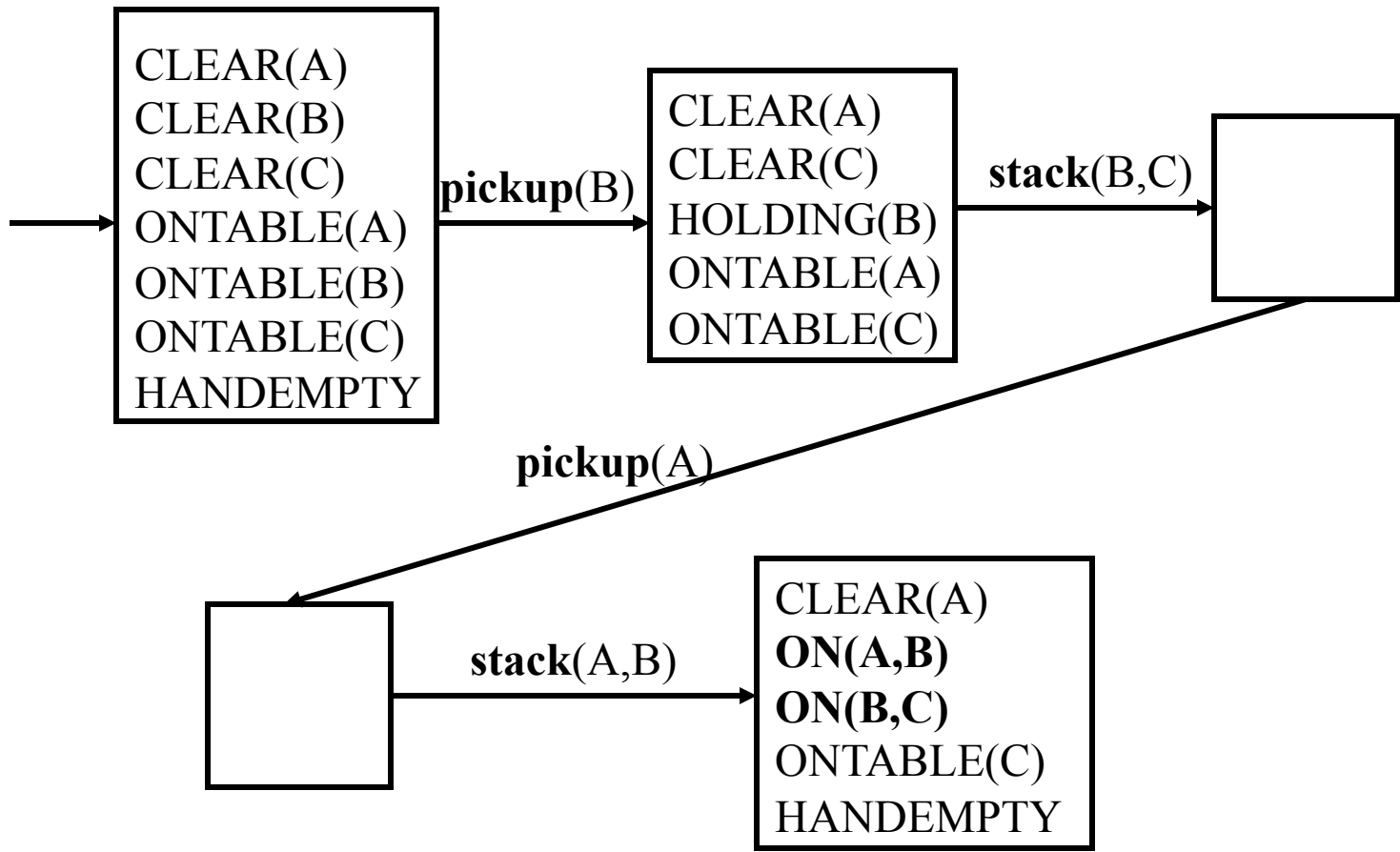
Goal Specification

ON(B,C) and ON(A,B)



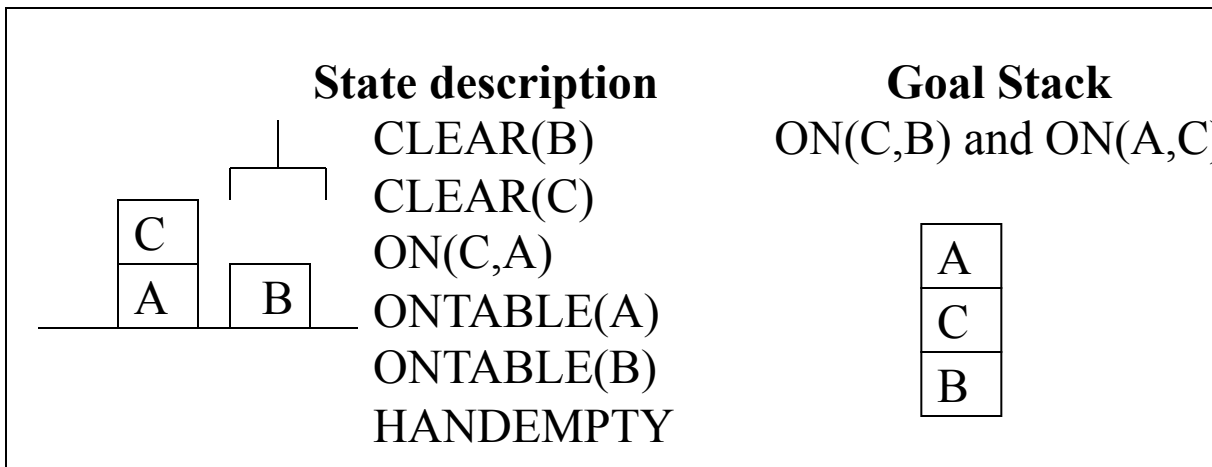
Example state that satisfies goal spec



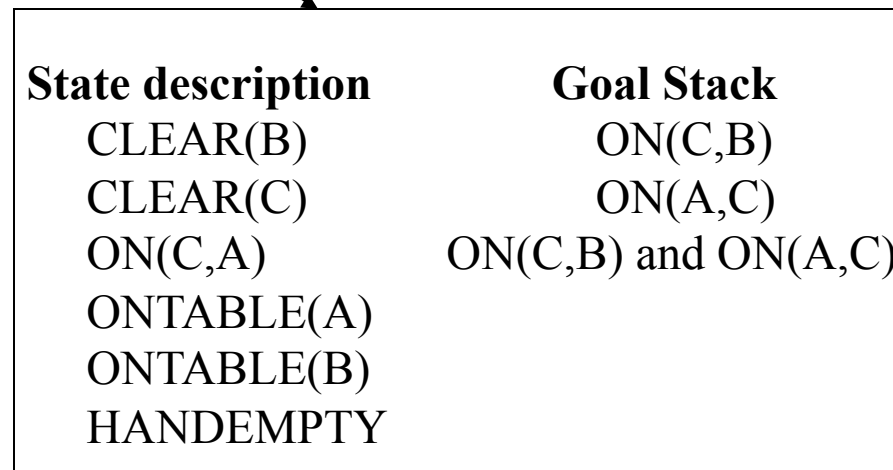
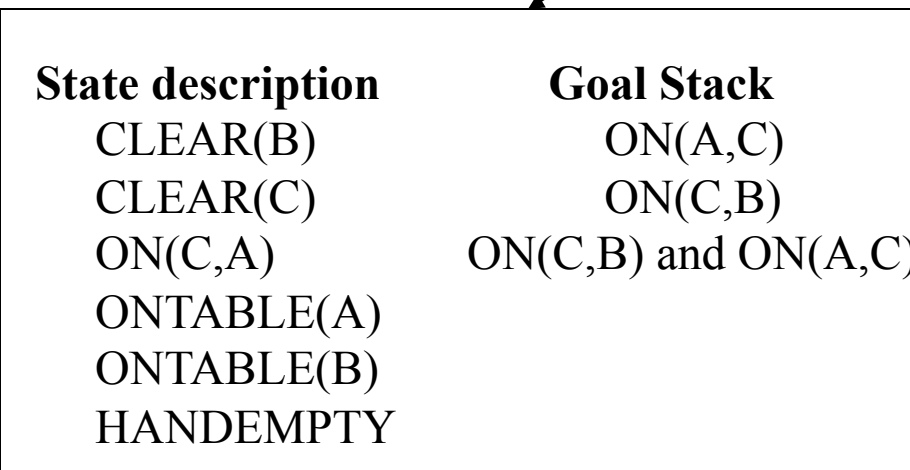


STRIPS

1

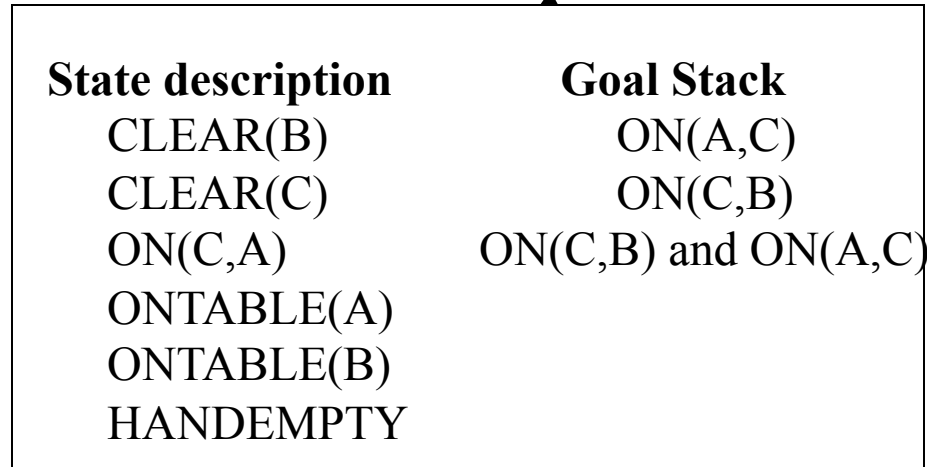


2

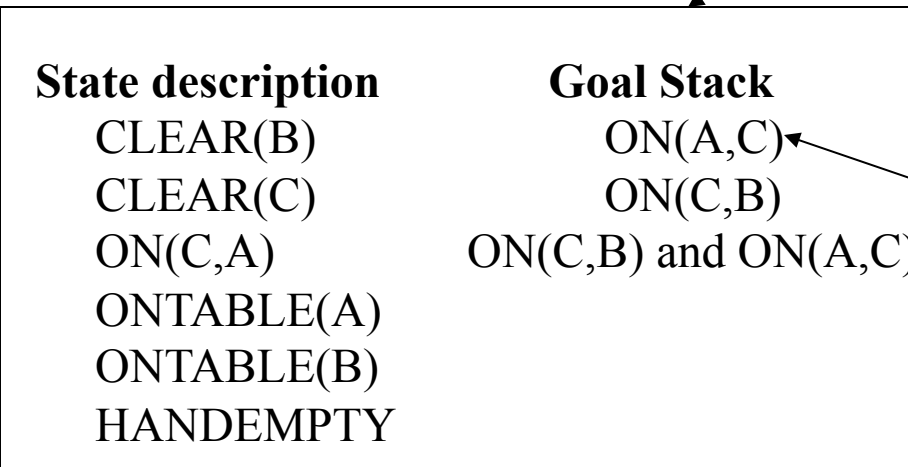


STRIPS

2



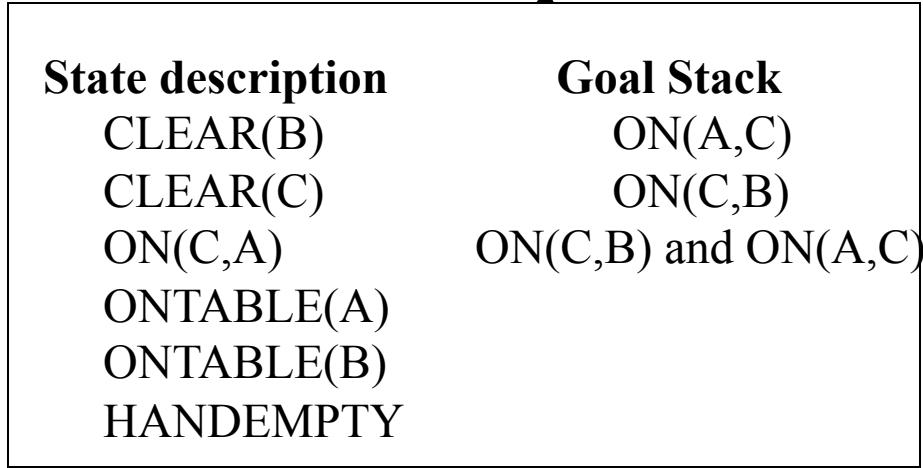
3



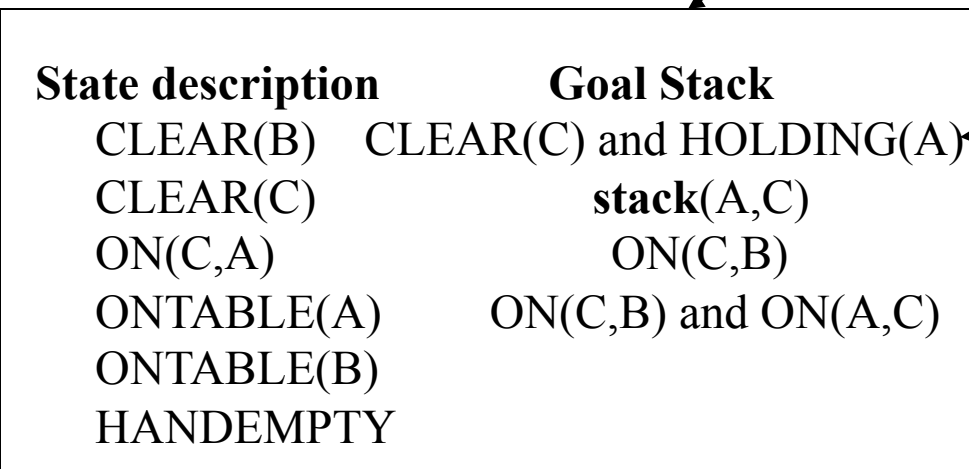
Stack(?x, ?y): PRE: HOLDING(?x)
CLEAR(?y)
{?x/A, ?y/C} DEL: HOLDING(?x)
CLEAR(?y)
ADD: HANDEEMPTY
ON(?x,?y)
CLEAR(?x)

STRIPS

2

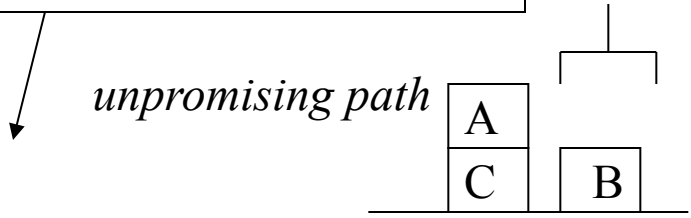


3



Stack(?x, ?y): PRE: HOLDING(?x)
 CLEAR(?y)
 DEL: HOLDING(?x)
 CLEAR(?y)
 ADD: HANDEEMPTY
 ON(?x,?y)
 CLEAR(?x)

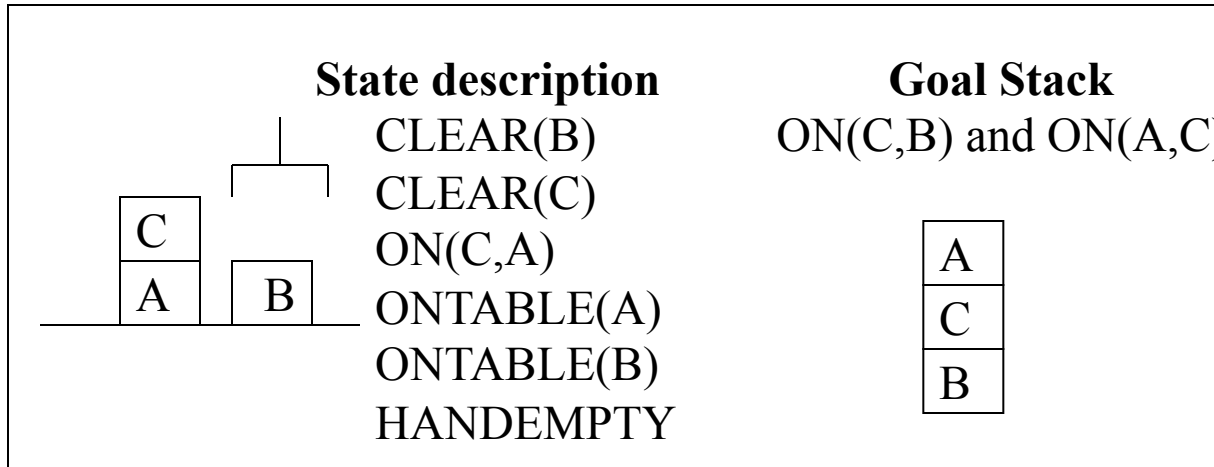
{?x/A, ?y/C}



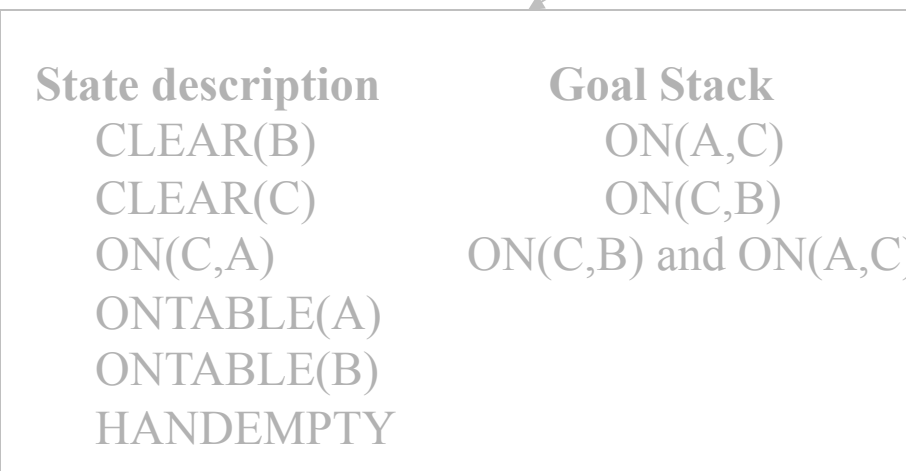
interacting subgoals: would have to undo ON(A,C) to achieve ON(C,B)

STRIPS

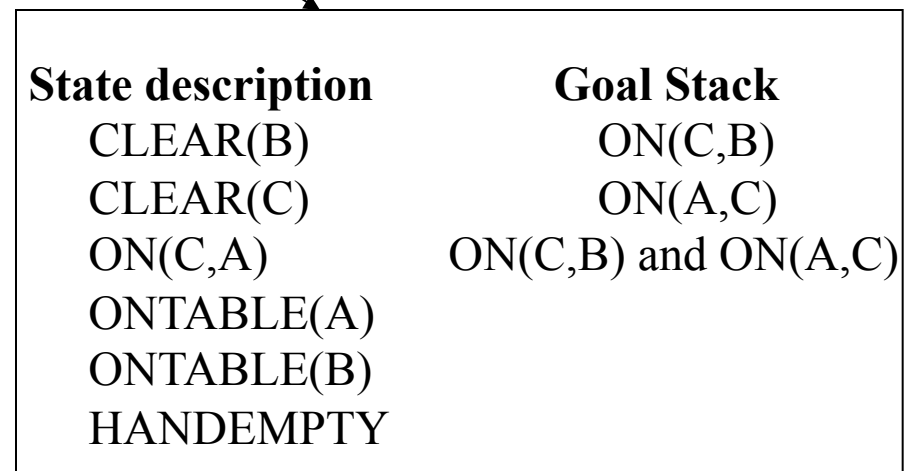
1

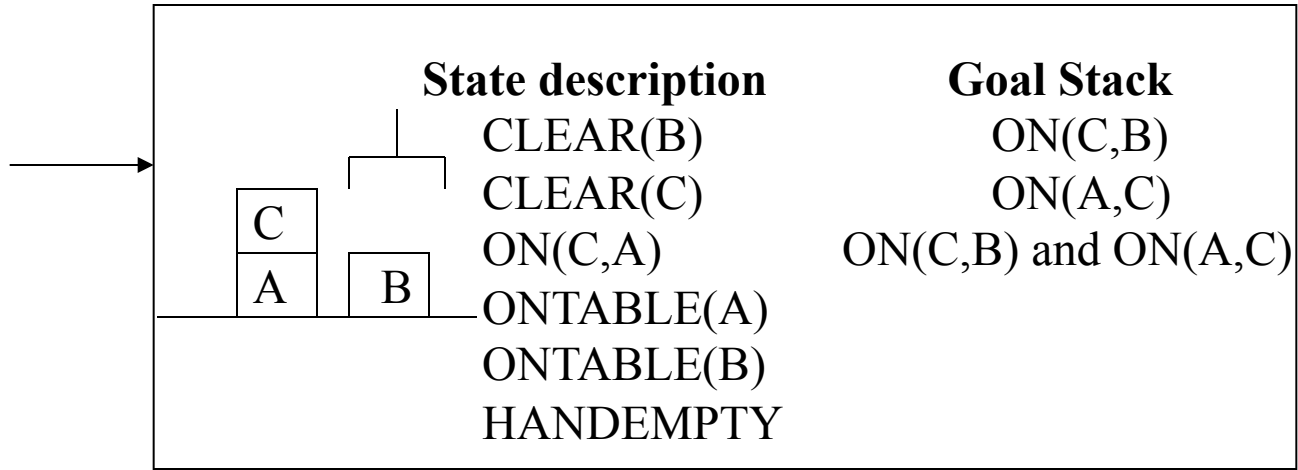


2



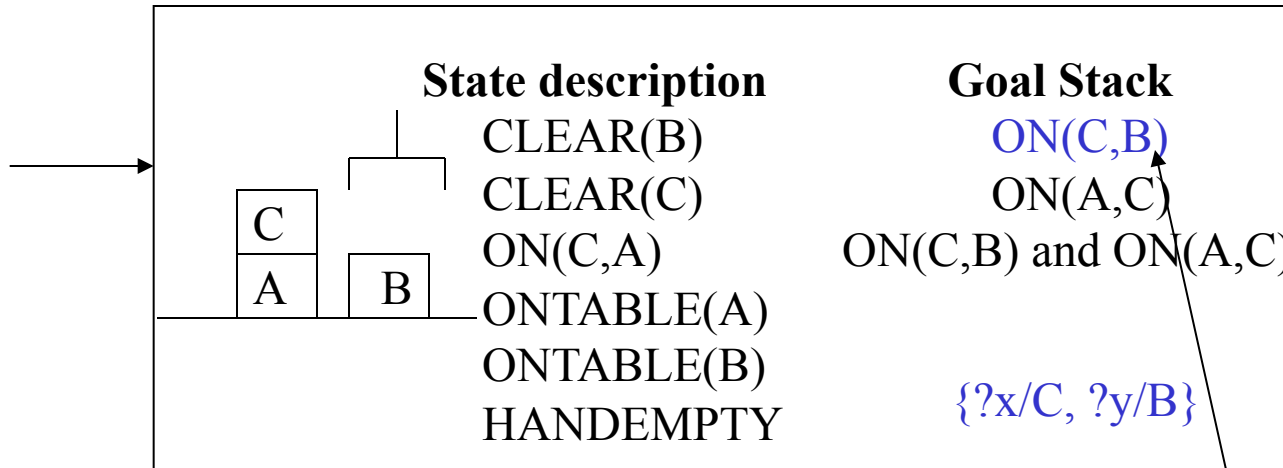
K



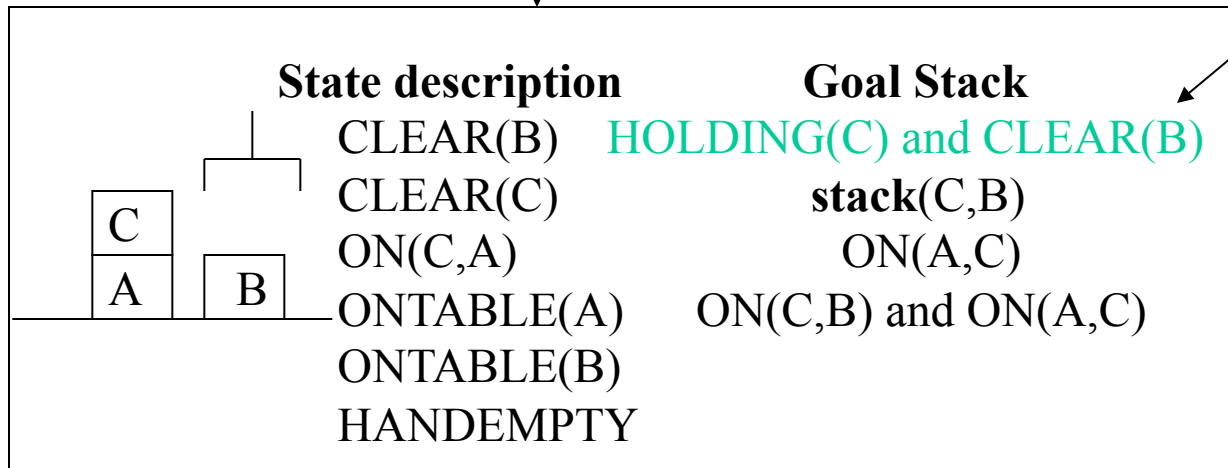


STRIPS

K

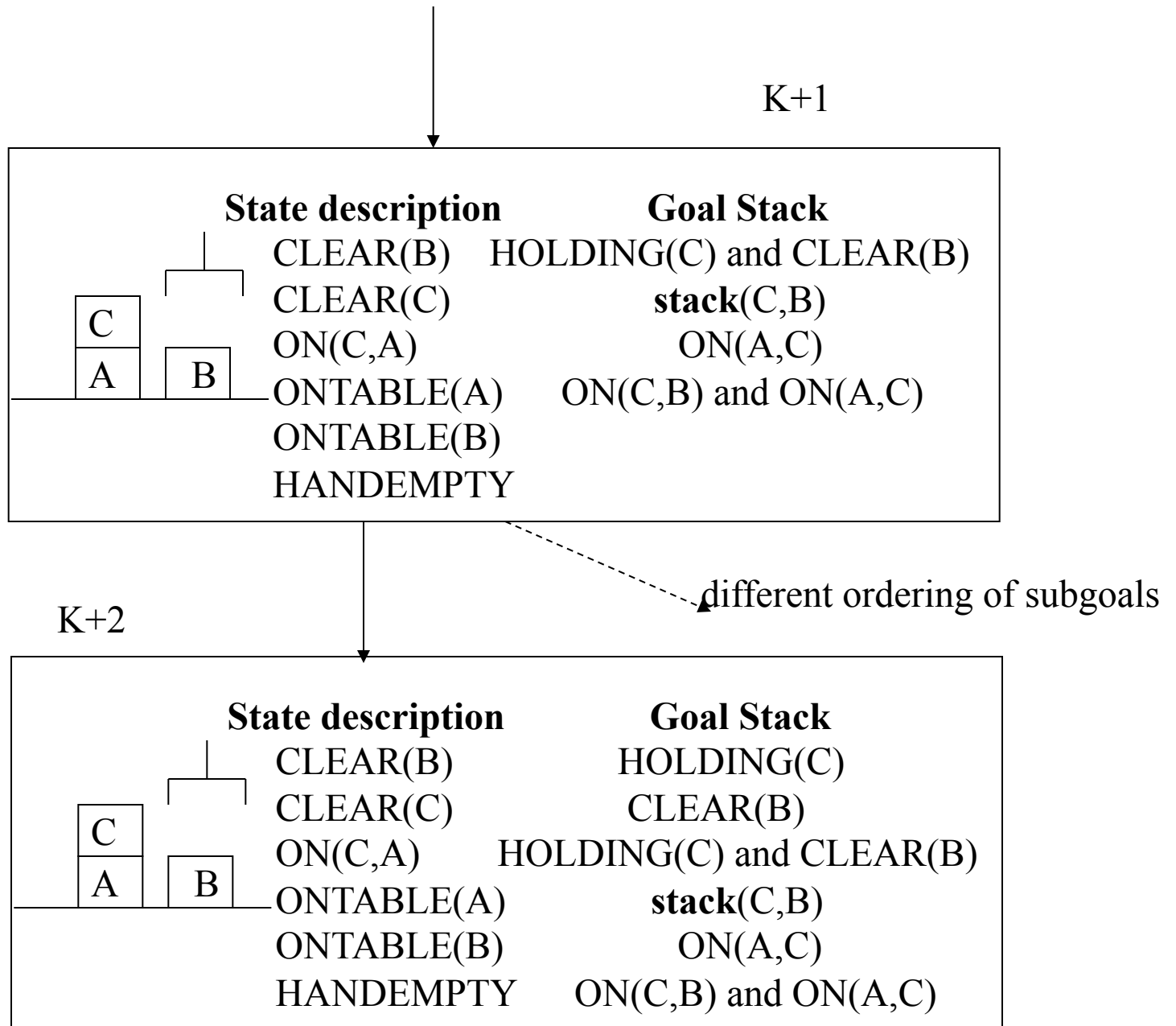


K+1



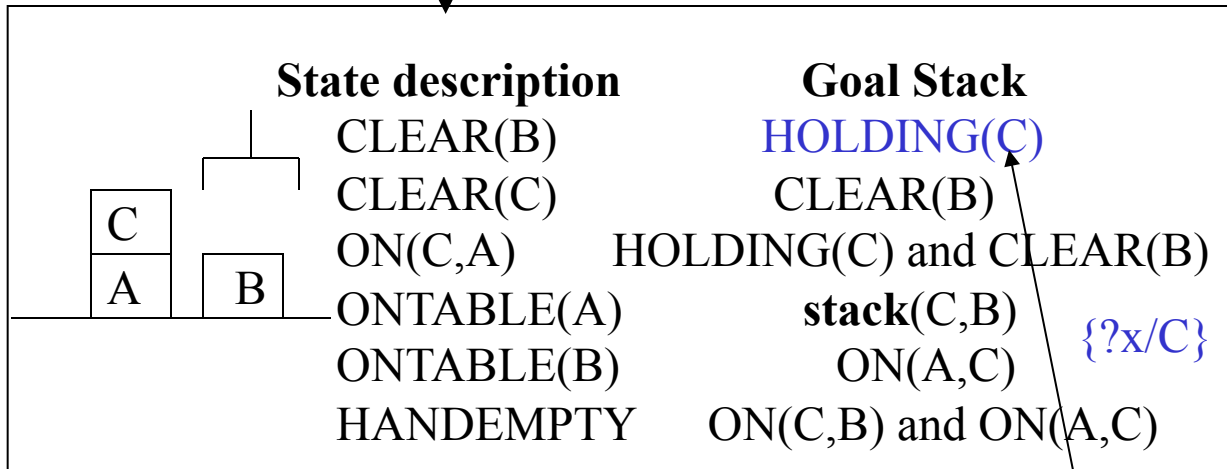
Stack(?x,?y) PRE: HOLD(?x),CLEAR(?y) DEL:....
ADD: ... ON(?x,?y)

STRIPS



STRIPS

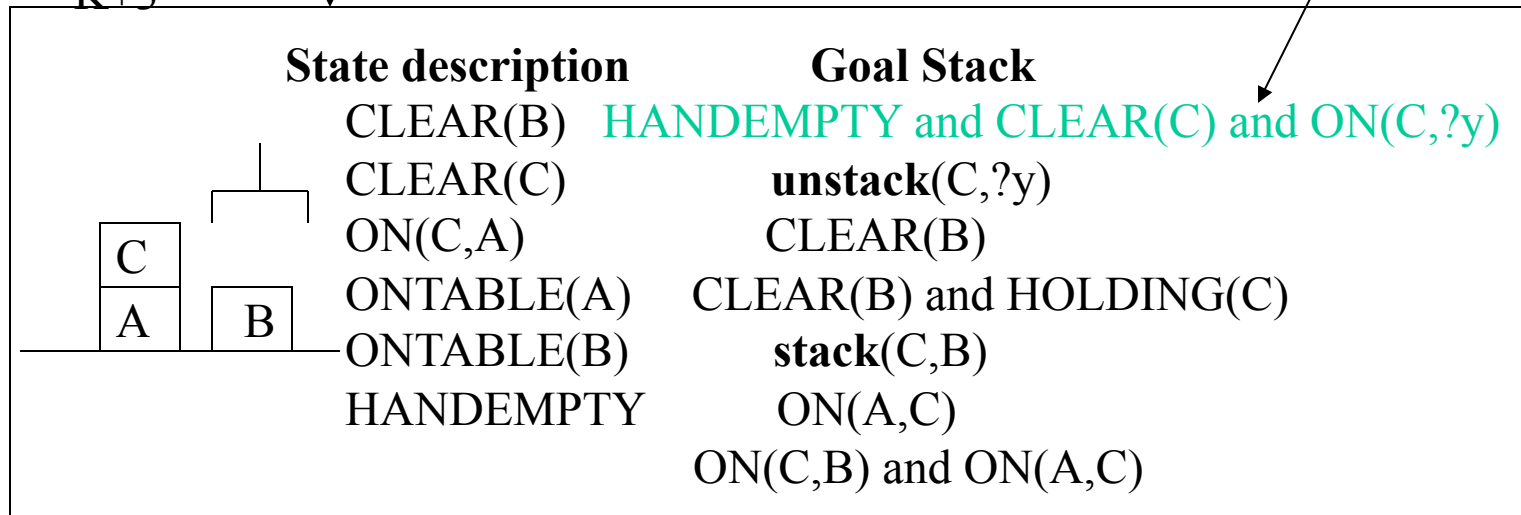
K+2



Use pickup to achieve
HOLDING(C)

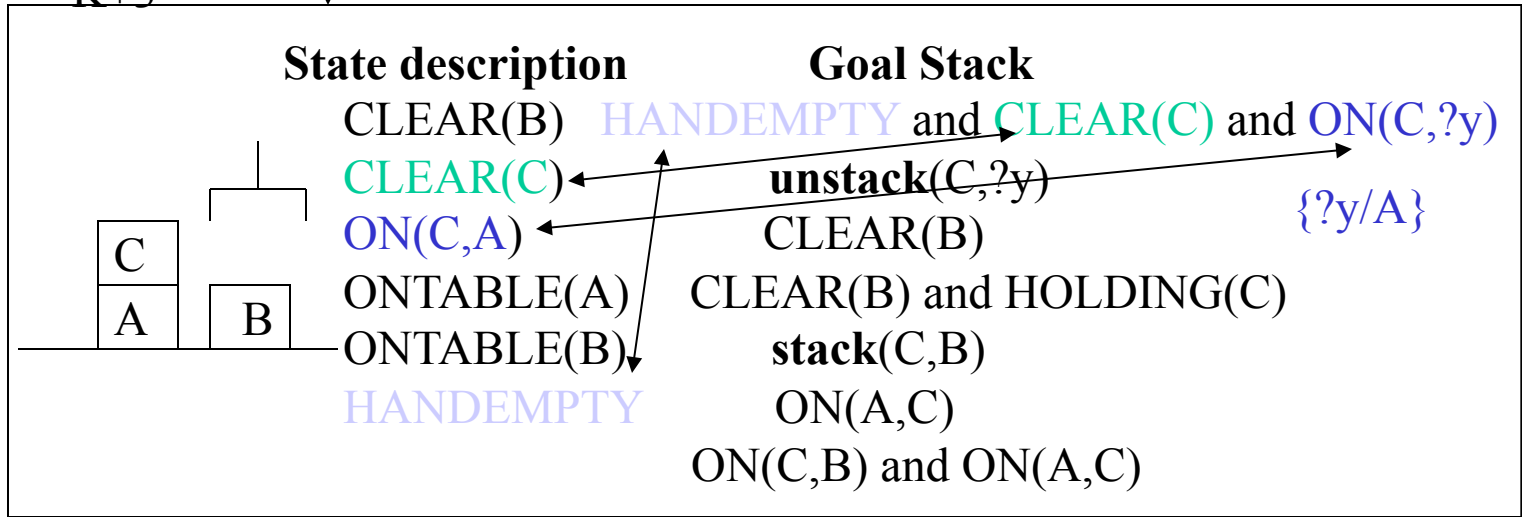
unstack(?x, ?y) PRE: HANDEEMPTY, CLEAR(?x), ON(?x,?y)
DEL:
ADD: HOLDING(?x), CLEAR(?y)

K+3



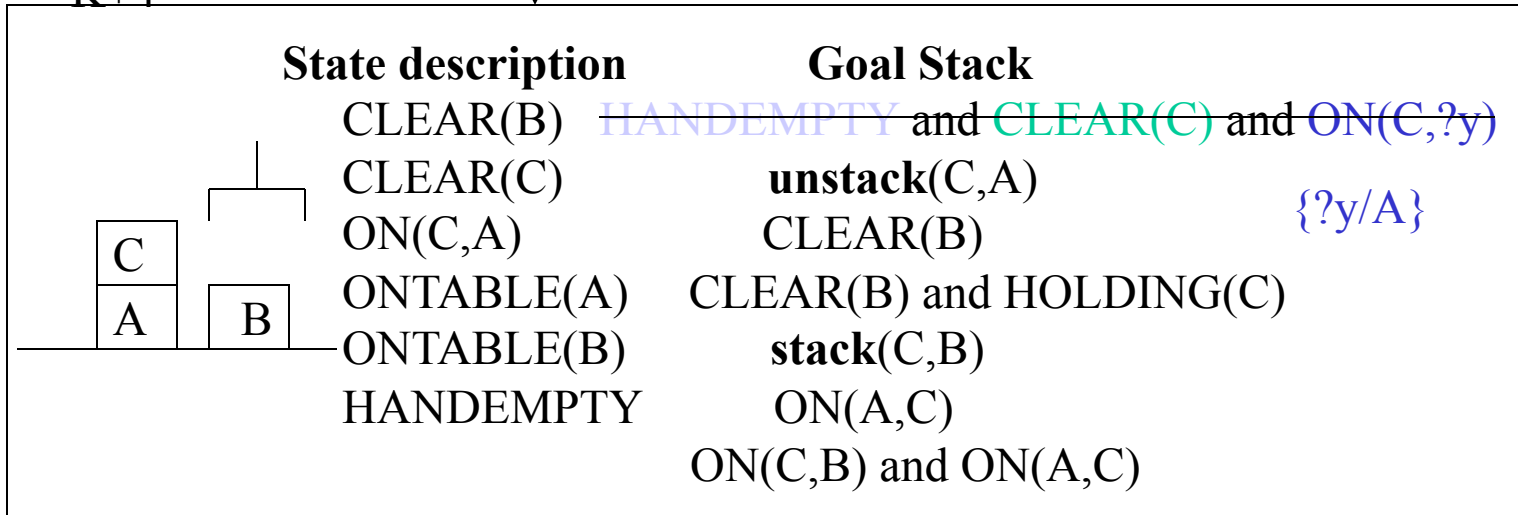
STRIPS

K+3



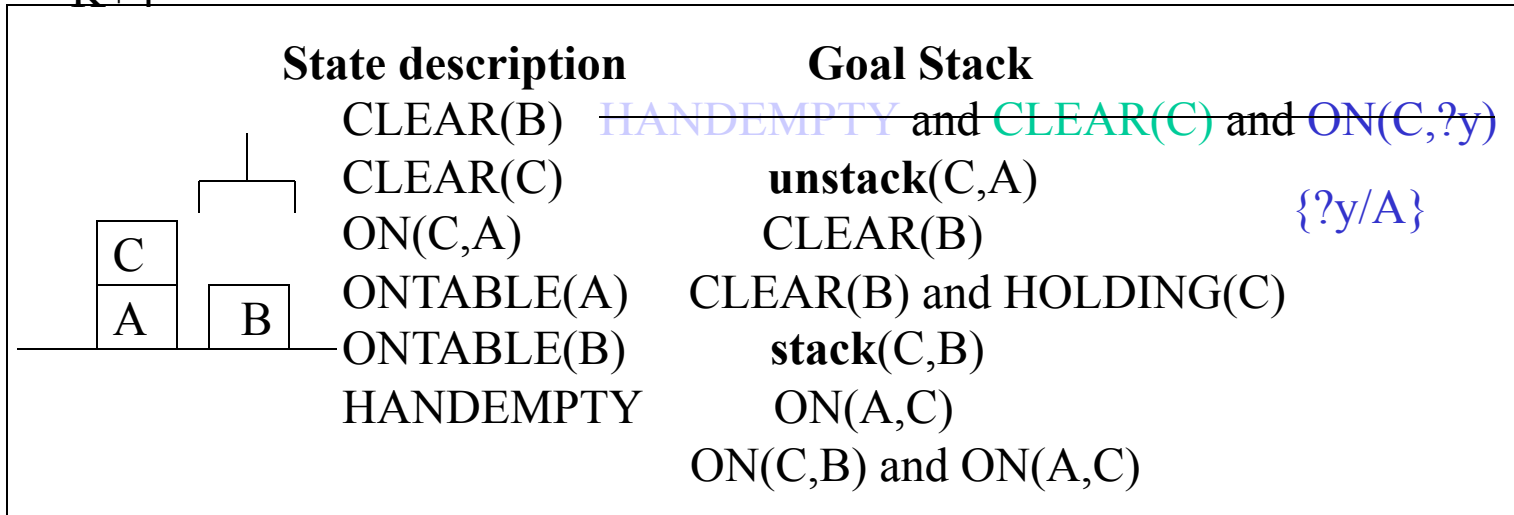
Pop satisfied subgoal

K+4



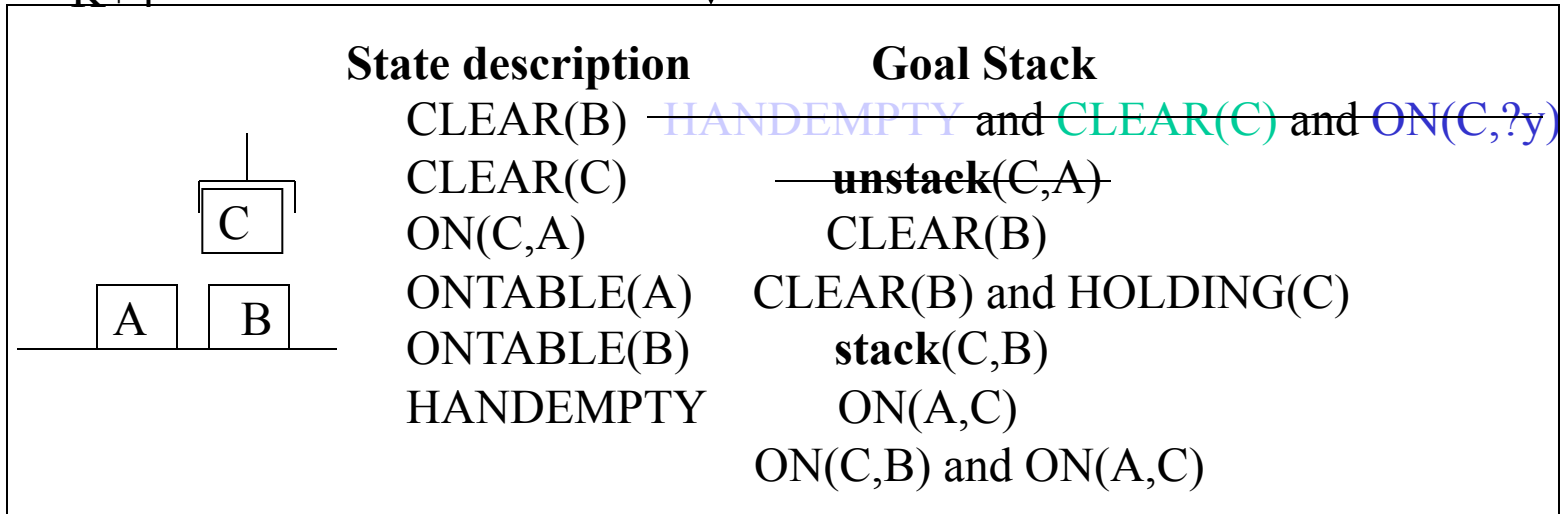
STRIPS

K+4



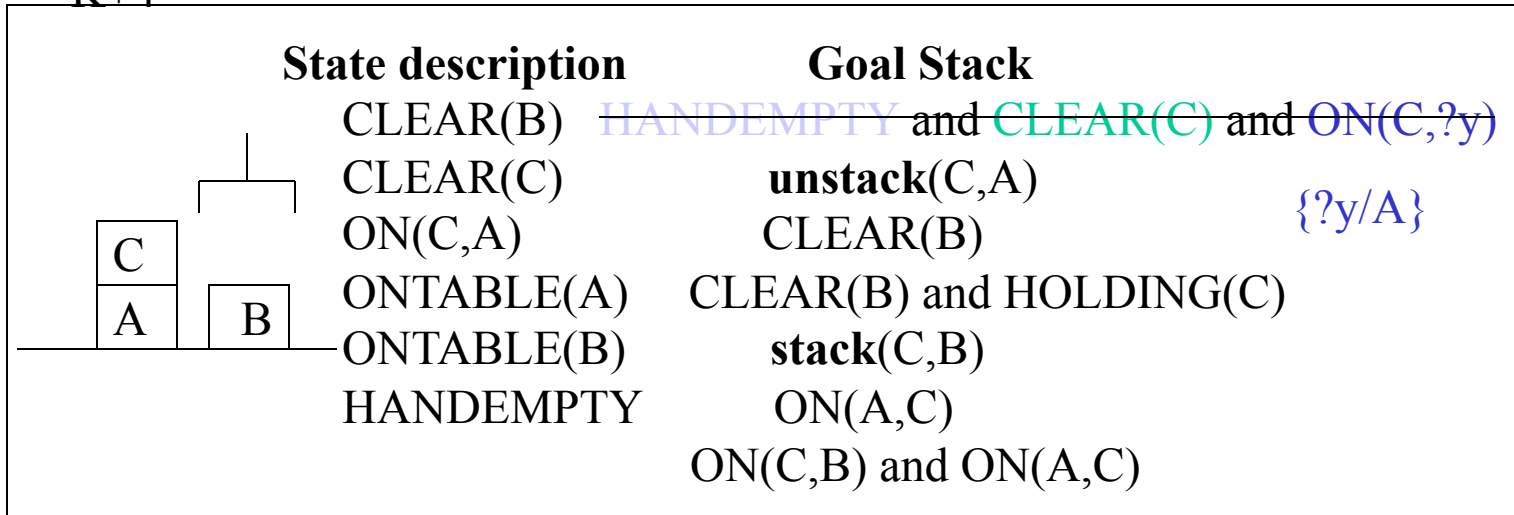
Apply **unstack(C,A)**

K+4



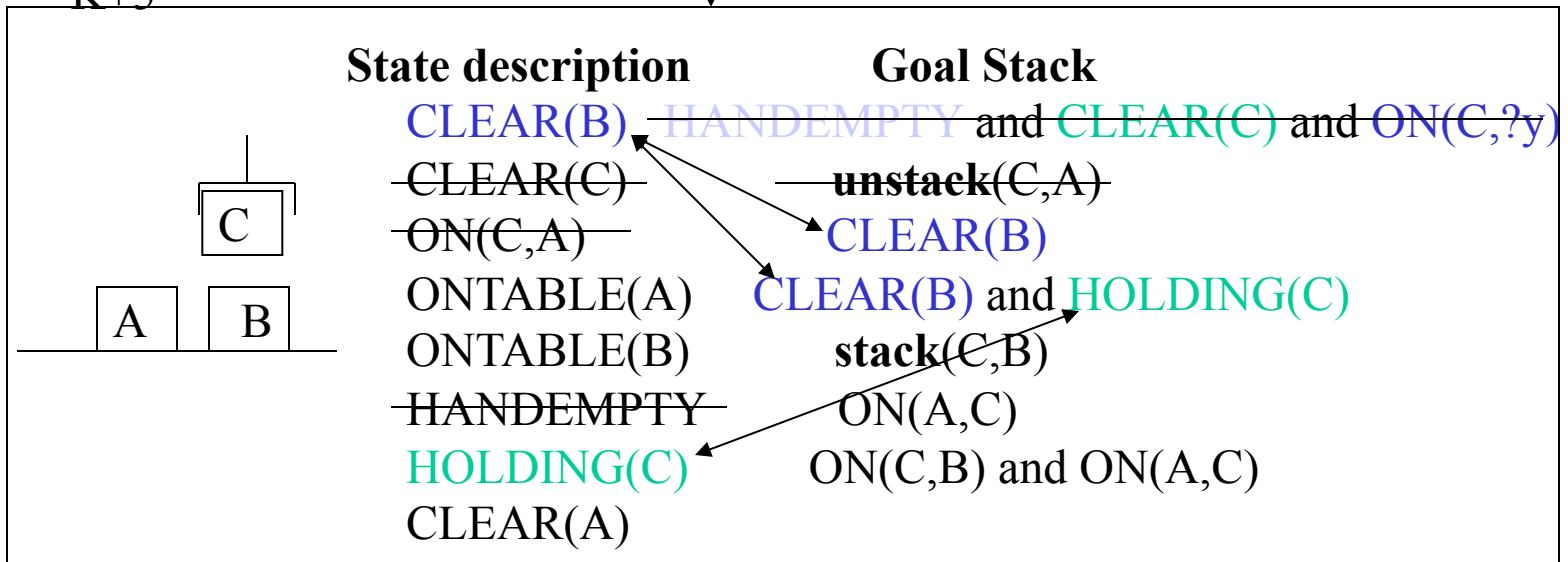
STRIPS

K+4



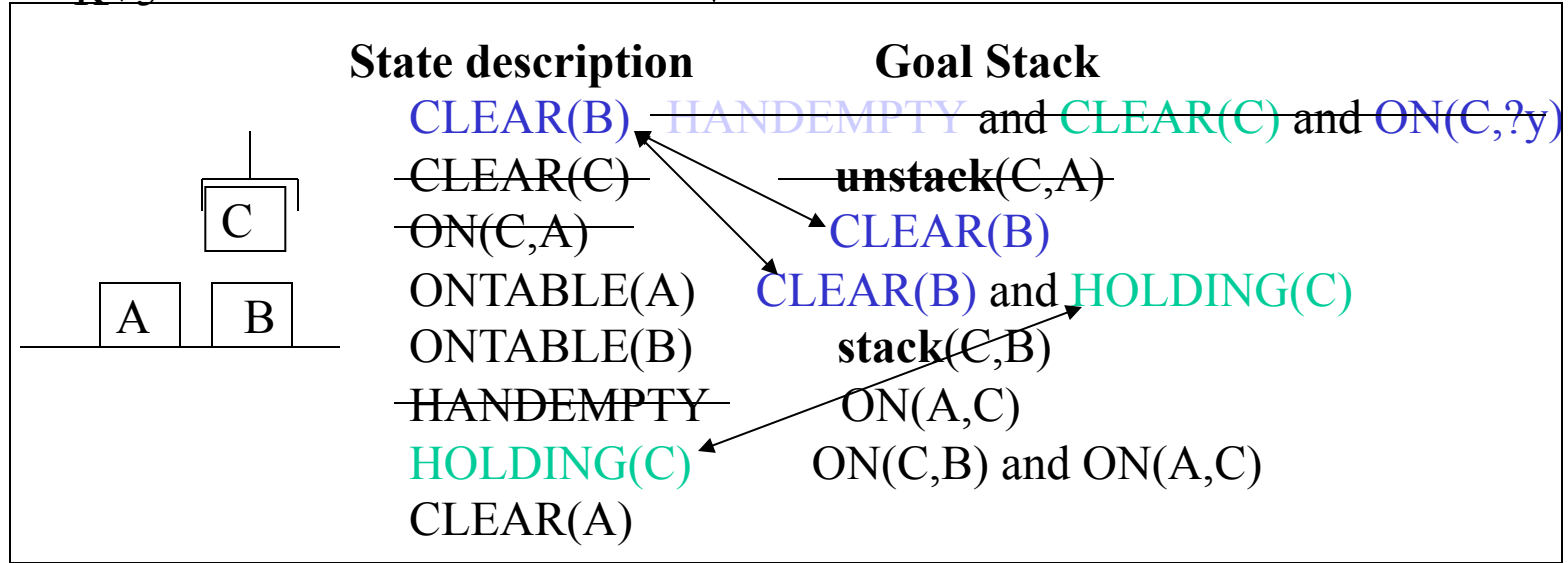
Apply **unstack(C,A)**

K+5



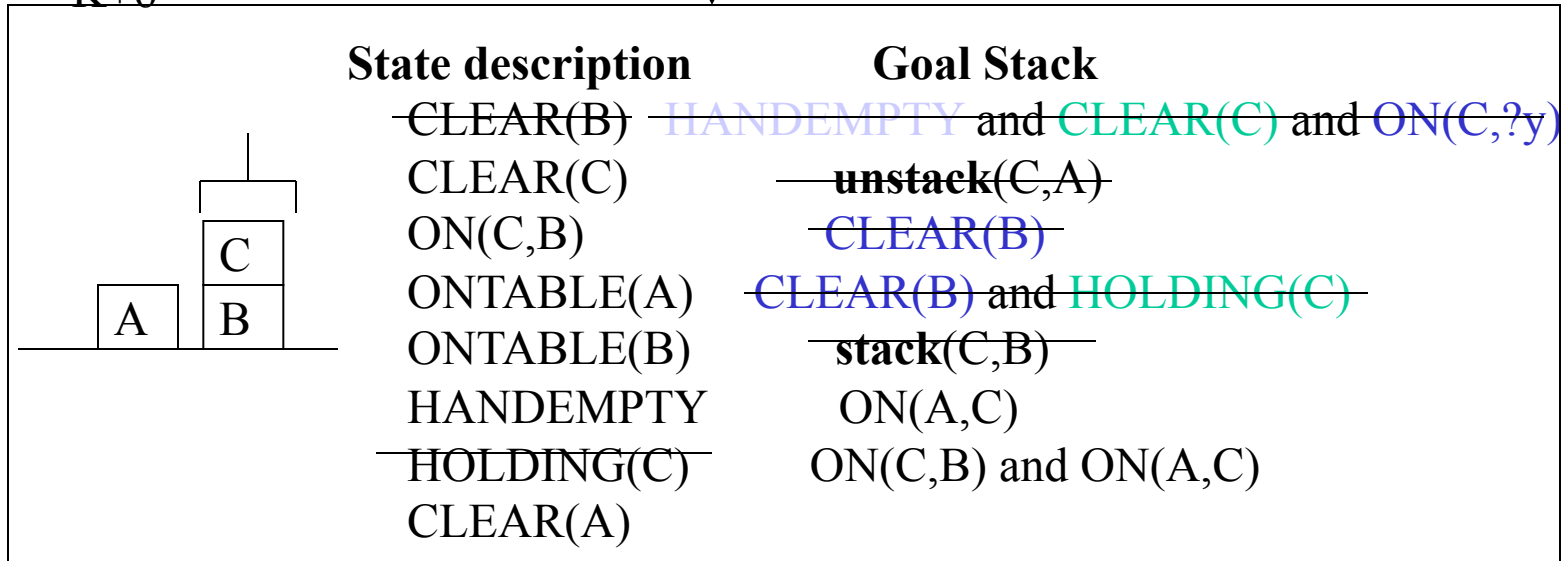
STRIPS

K+5



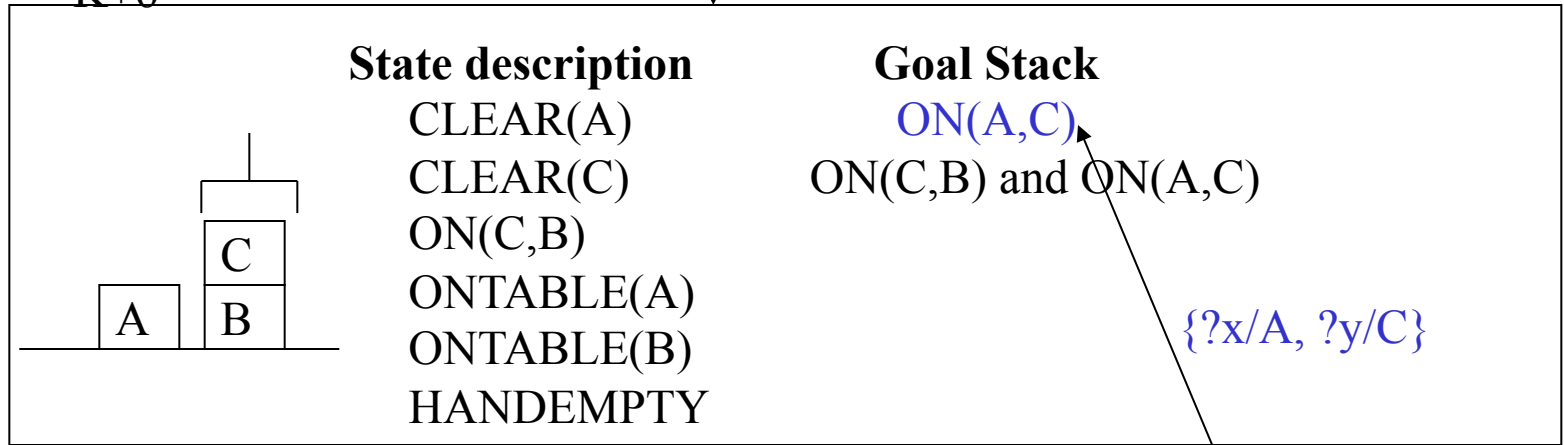
Pop goals and apply **stack(C,B)**

K+6



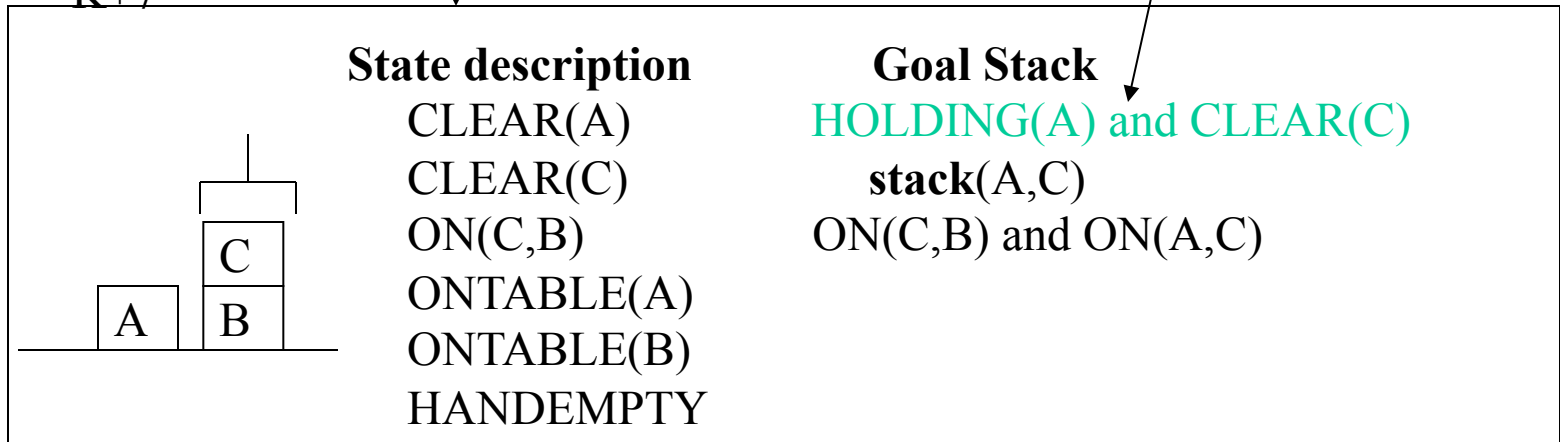
STRIPS

K+6

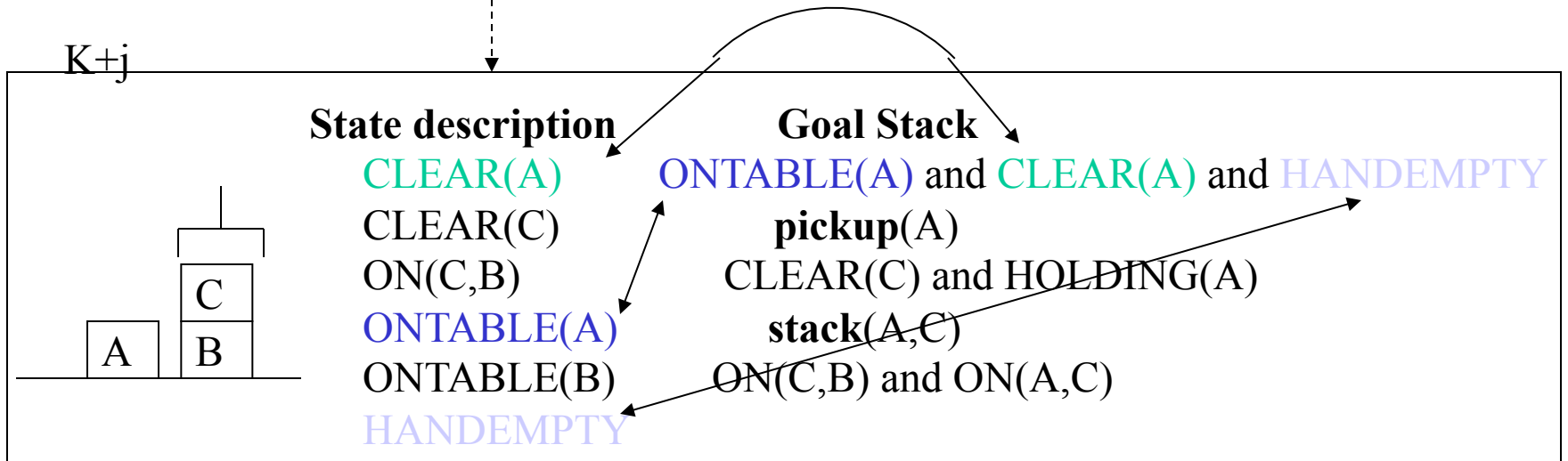
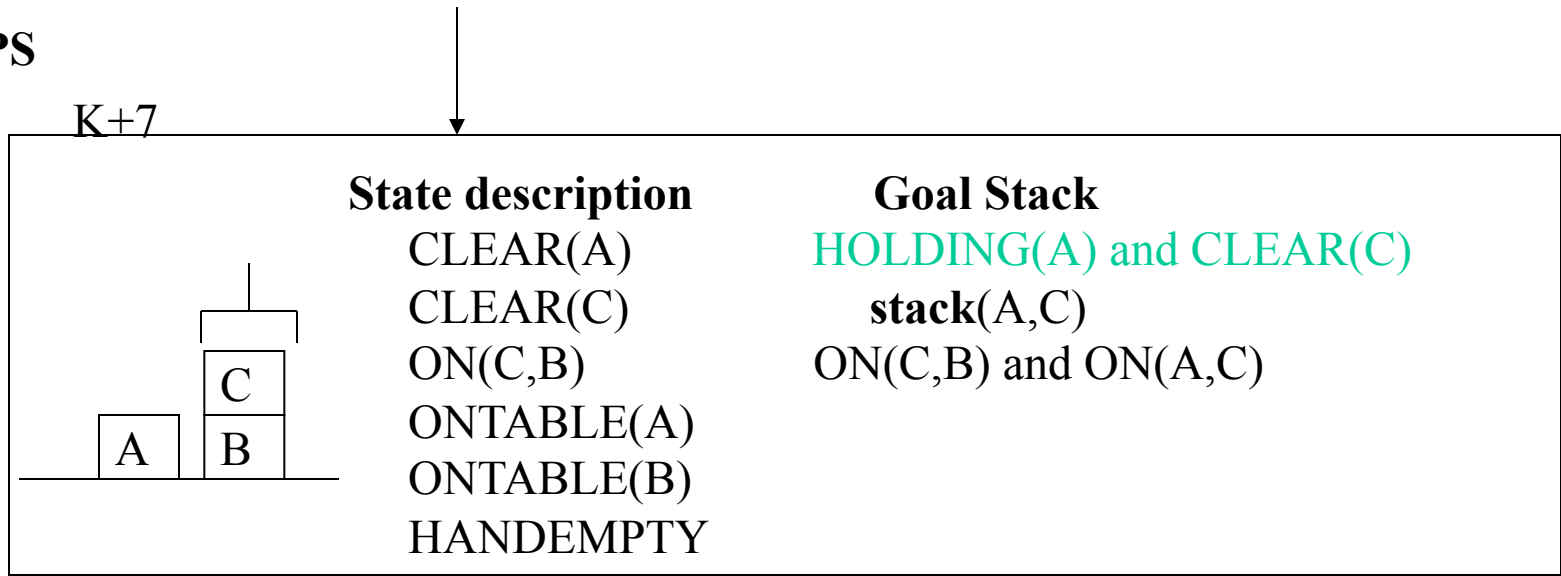


Stack(?x,?y) PRE: HOLDING(?x), CLEAR(?y)
DEL:
ADD: HANDEEMPTY, ON(?x,?y), CLEAR(?x)

K+7



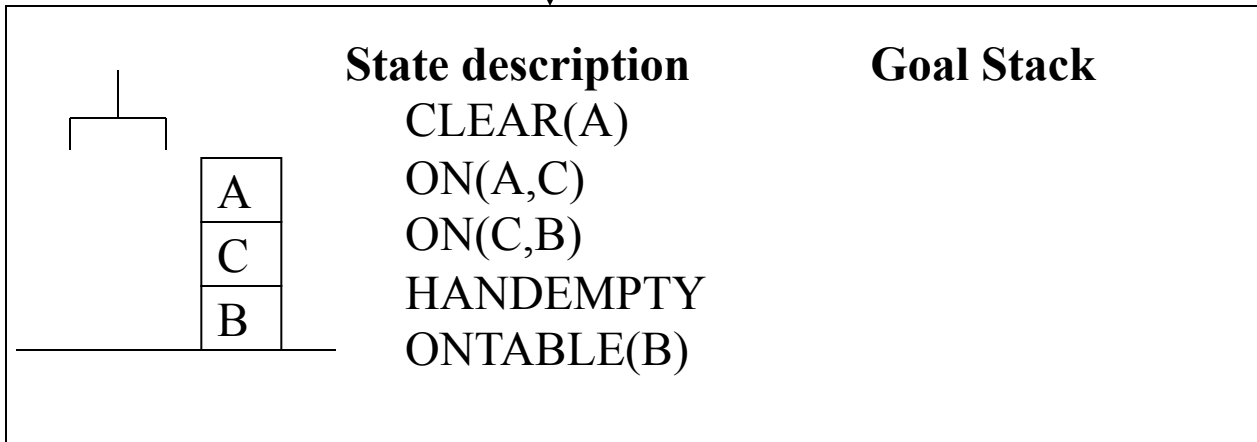
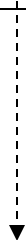
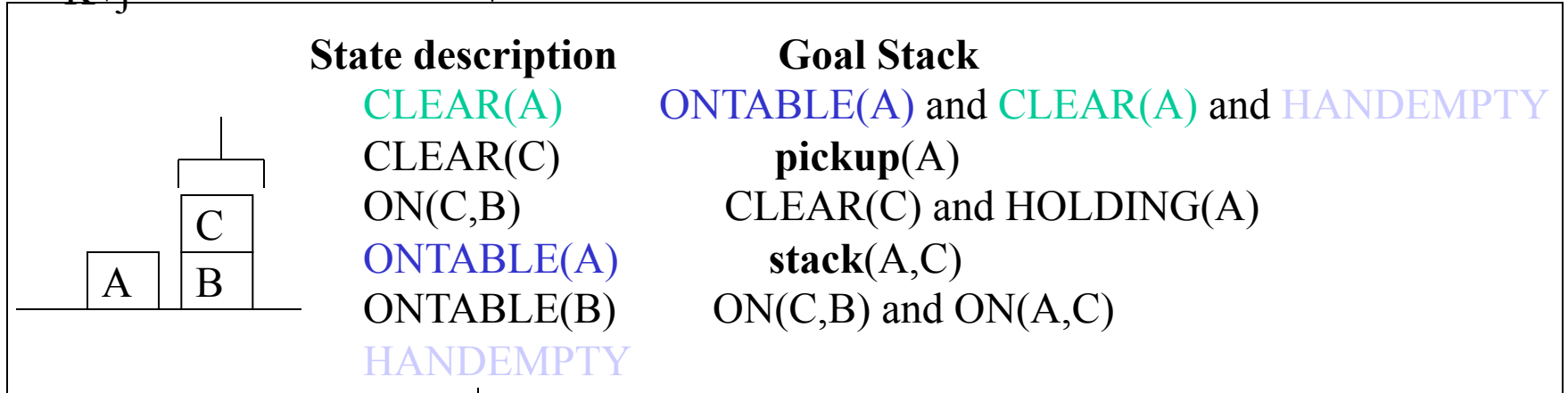
STRIPS



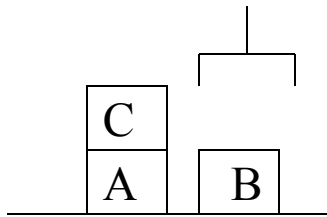
STRIPS



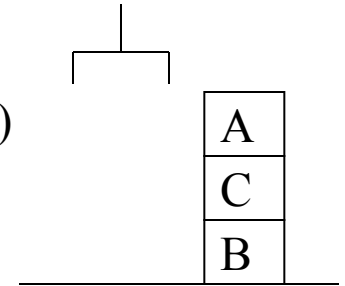
$K+j$



Final Plan given by order of “applied” operators:



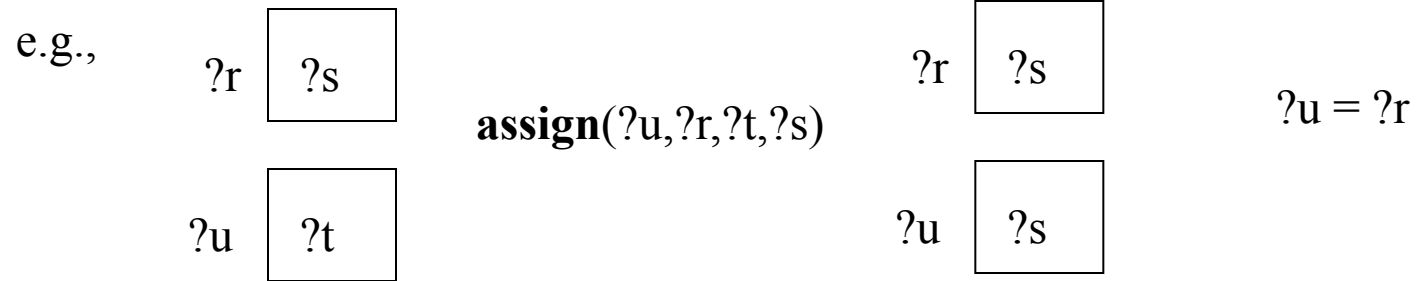
unstack(C,A) → stack(C,B) → pickup(A) → stack(A,C)



Think about:

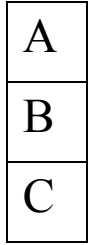
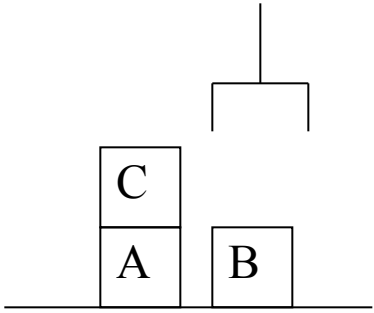
- 1) STRIPS search for a plan using Depth-First search. How does it know when to abandon a path?
- 2) Fill in the steps of my example where I have -----> transitions.
- 3) Use STRIPS to solve InitialState = {CLEAR(B), ON(C,A), CLEAR(C), ONTABLE(A), ONTABLE(B), HANDEEMPTY}
GoalSpec = {ON(A,B), ON(B,C)}
- 4) Use STRIPS to solve InitialState = {CLEAR(C), ON(A,B), ONTABLE(C), ONTABLE(B), CLEAR(A), HANDEEMPTY}
GoalSpec = {ON(A,B), ON(B,C)}

5. Assume a single operator: **assign**(?u, ?r, ?t, ?s)
PRE: CONT(?r,?s), CONT(?u,?t)
DEL: CONT(?u,?t)
ADD: CONT(?u,?s)



Use STRIPS to solve InitialState={CONT(V1, a), CONT(V2, b), CONT(V3, c)}
GoalSpec = {CONT(V1, b), CONT(V2, a)}

Pop : the following slides represent *one* path in a search for a plan



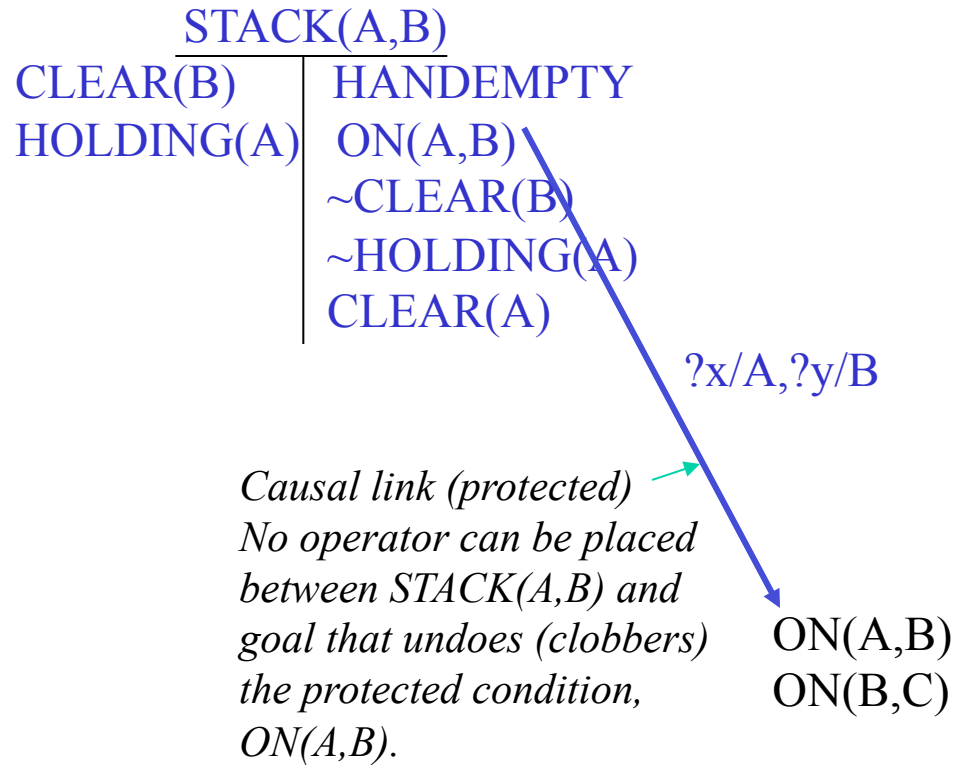
ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEMPTY

ON(A,B)
ON(B,C)

Initial State

Goal Spec

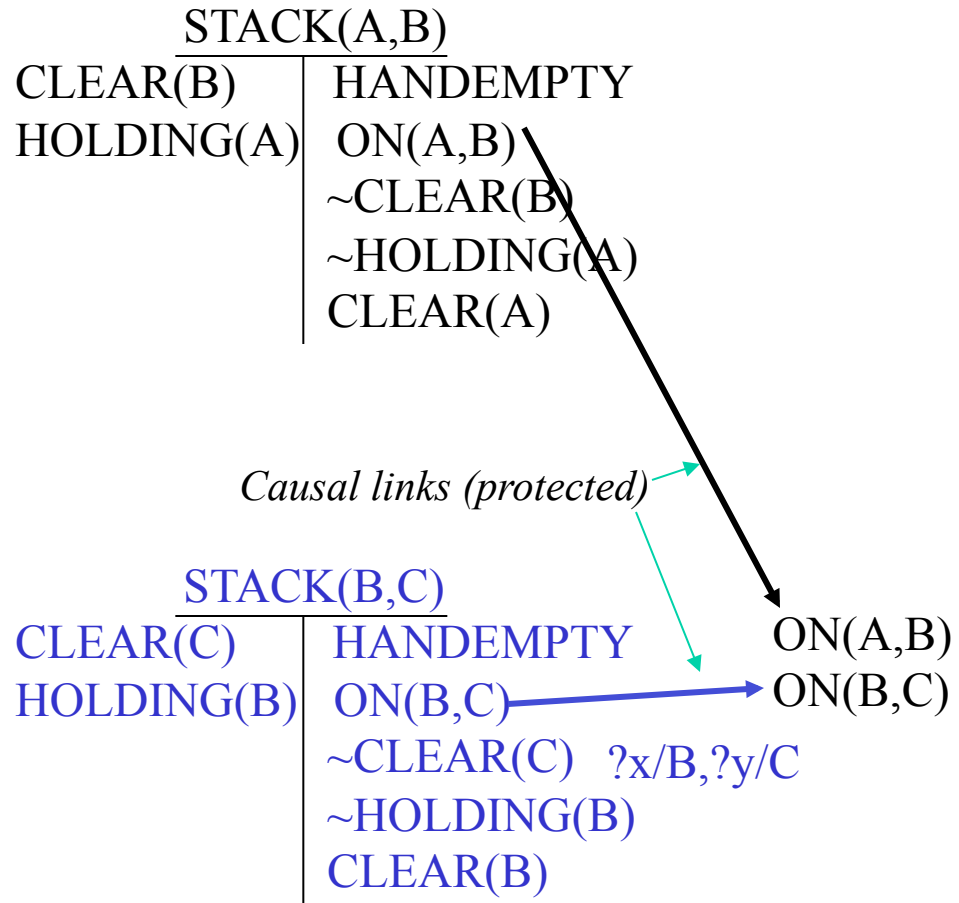
Tweak



ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEEMPTY

Step addition

Tweak



ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEEMPTY

Step addition

Tweak

STACK(A,B)

CLEAR(B)	HANDEEMPTY
HOLDING(A)	ON(A,B)
	~CLEAR(B)
	~HOLDING(A)
	CLEAR(A)

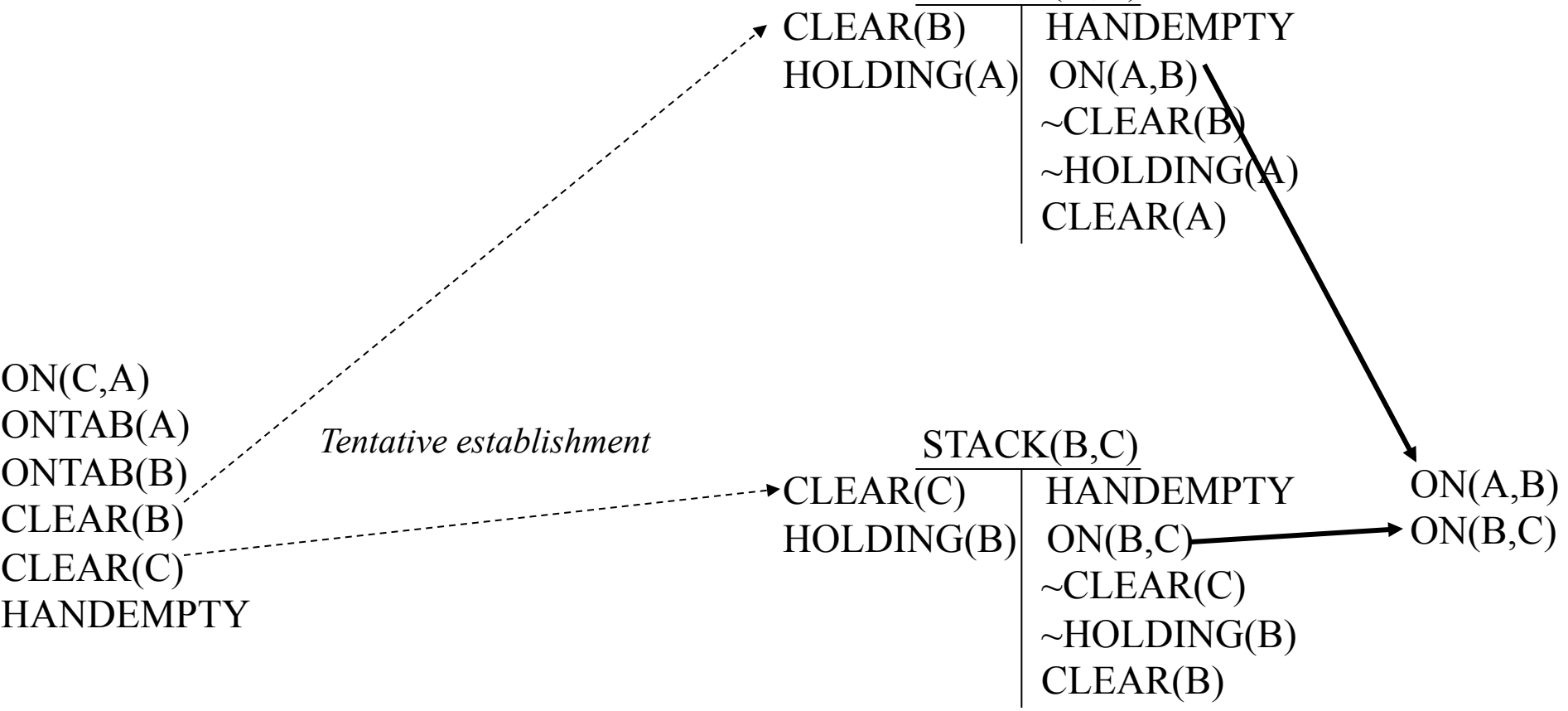
STACK(B,C)

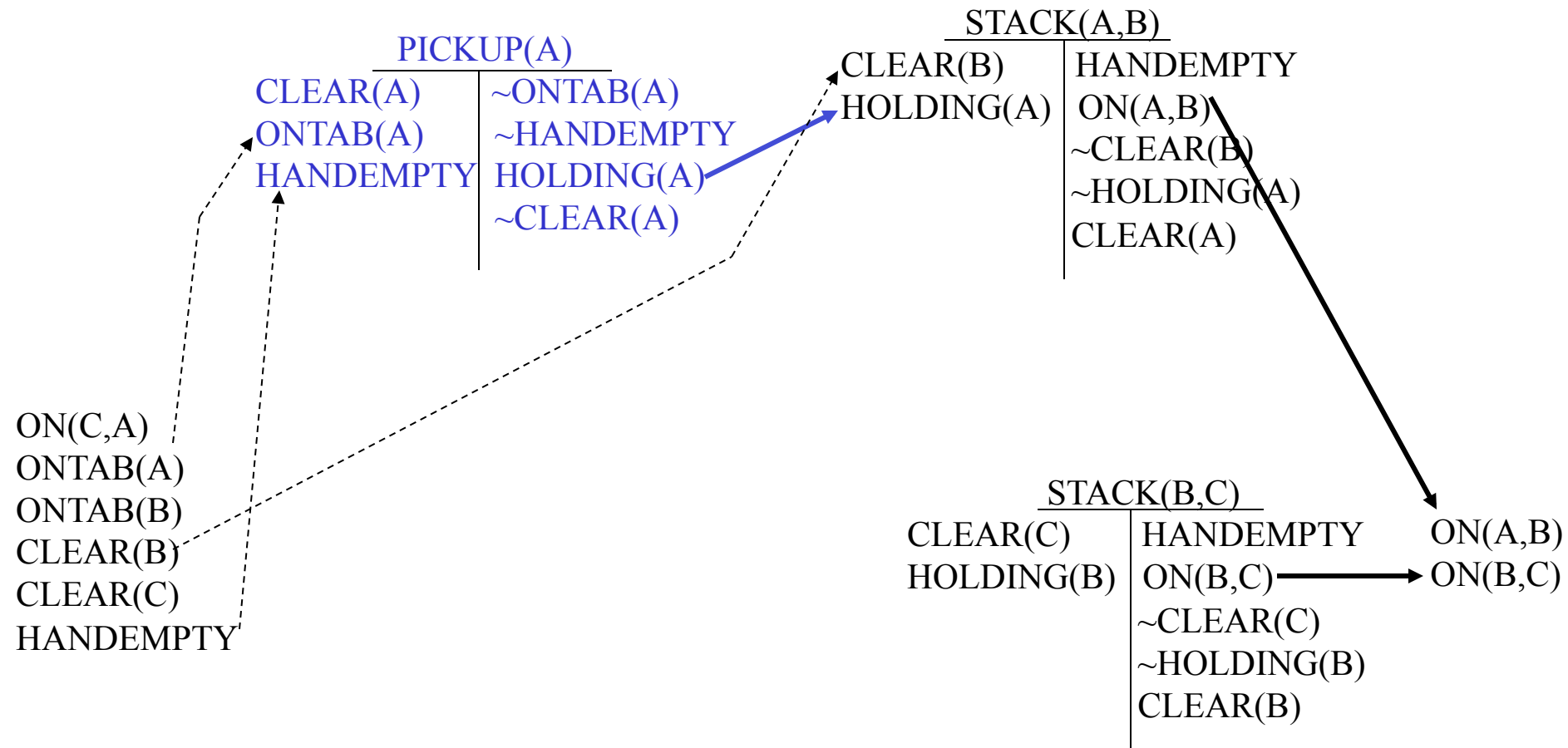
CLEAR(C)	HANDEEMPTY
HOLDING(B)	ON(B,C)
	~CLEAR(C)
	~HOLDING(B)
	CLEAR(B)

ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEEMPTY

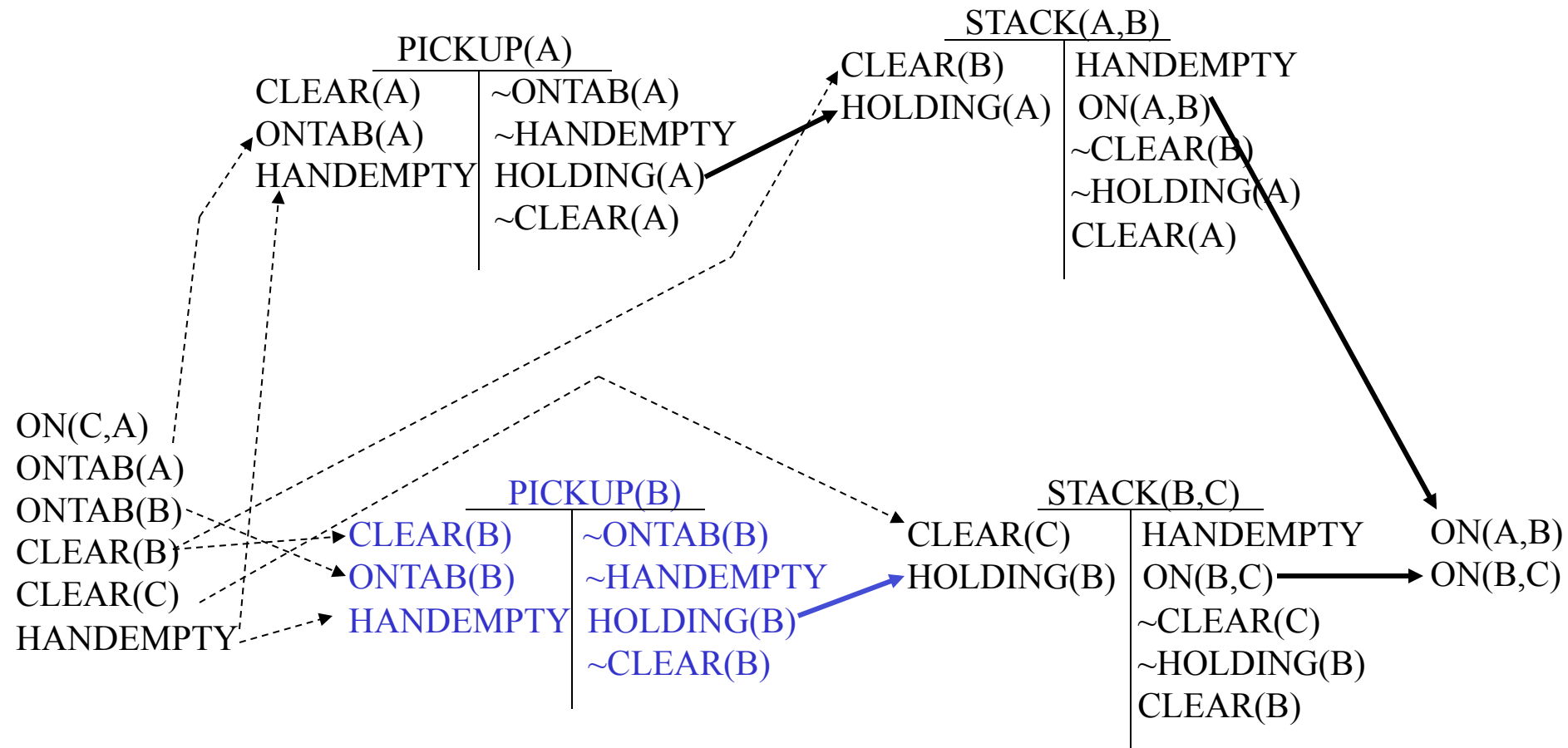
Tentative establishment

ON(A,B)
ON(B,C)

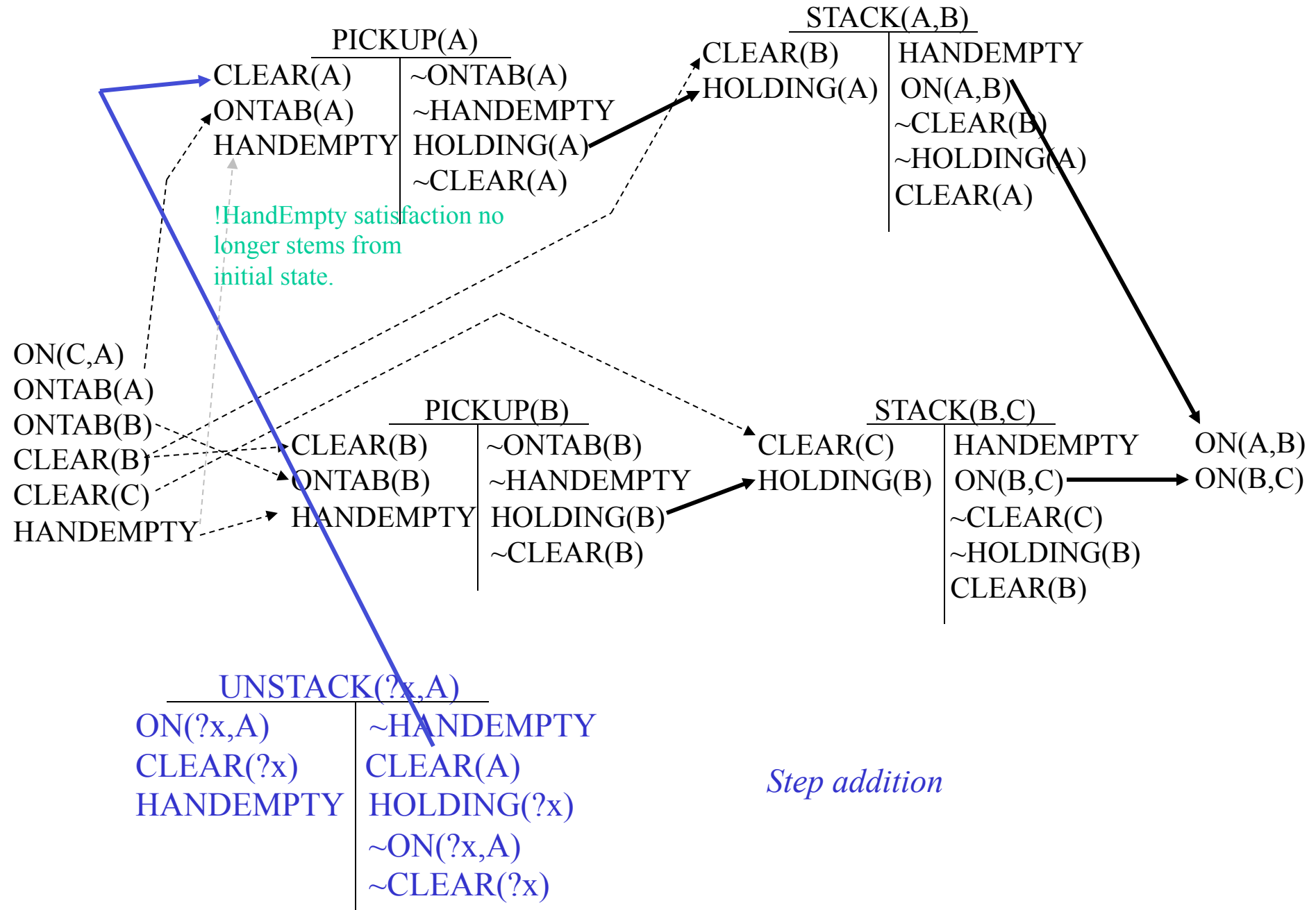


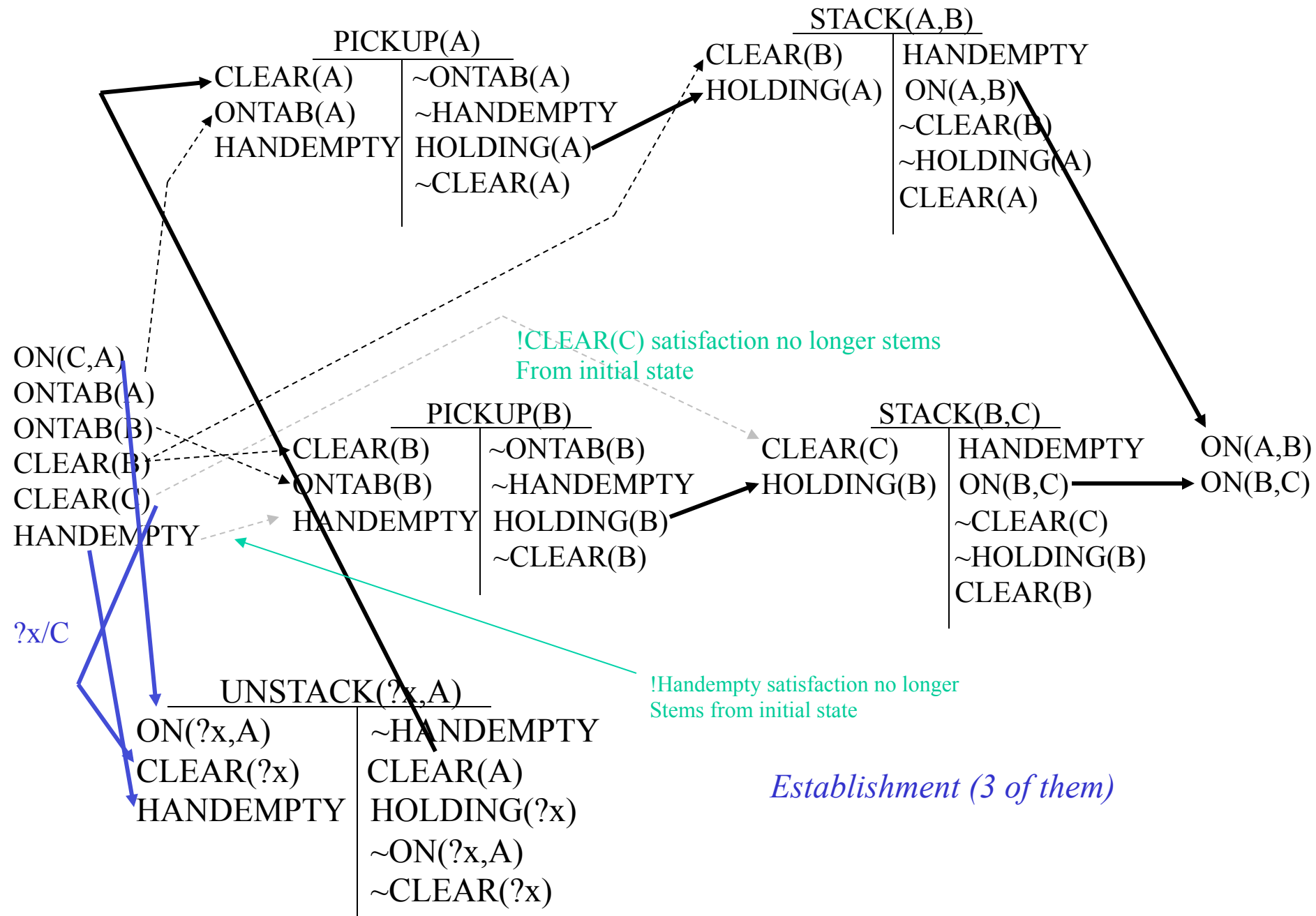


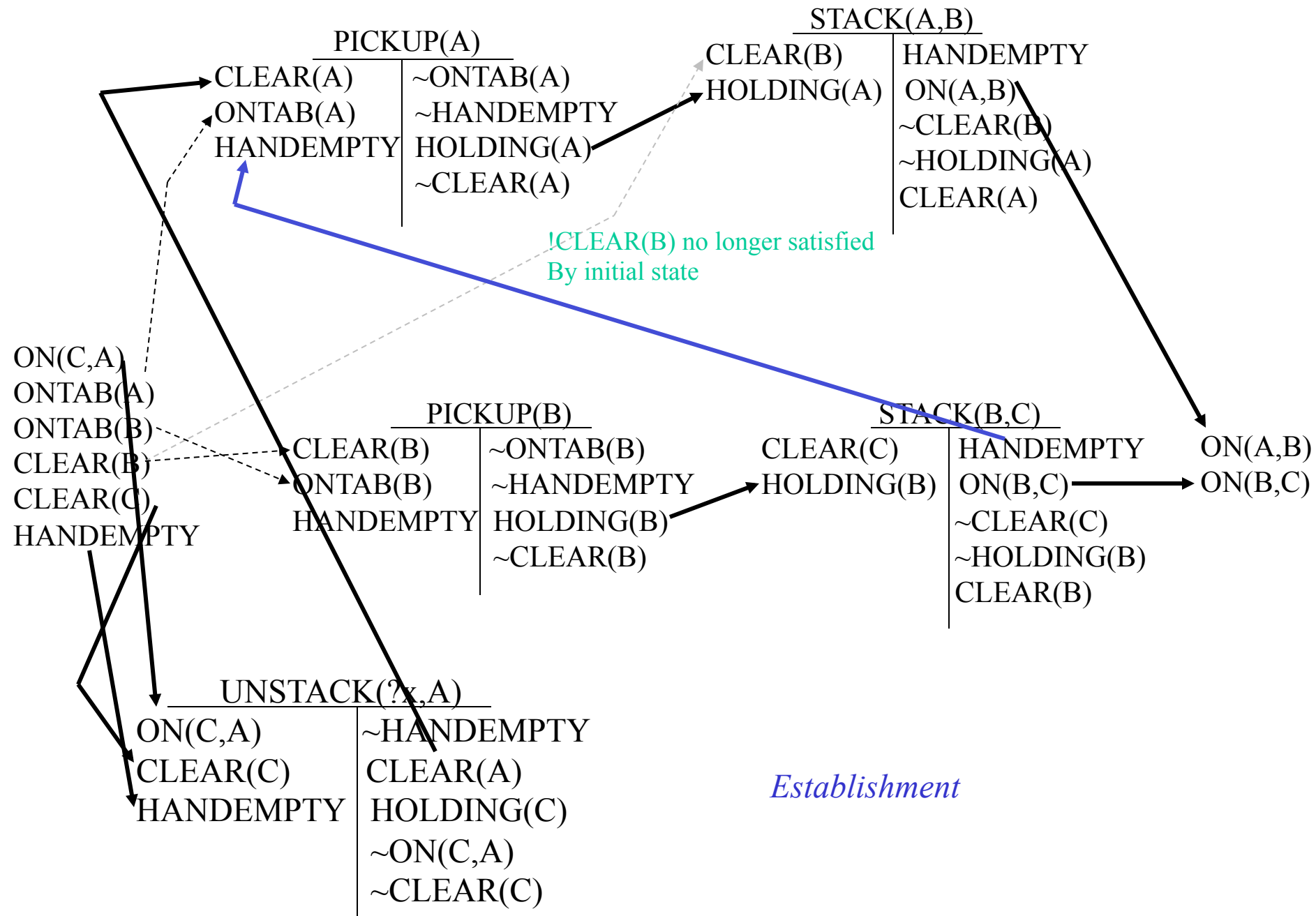
Step addition and tentative establishment

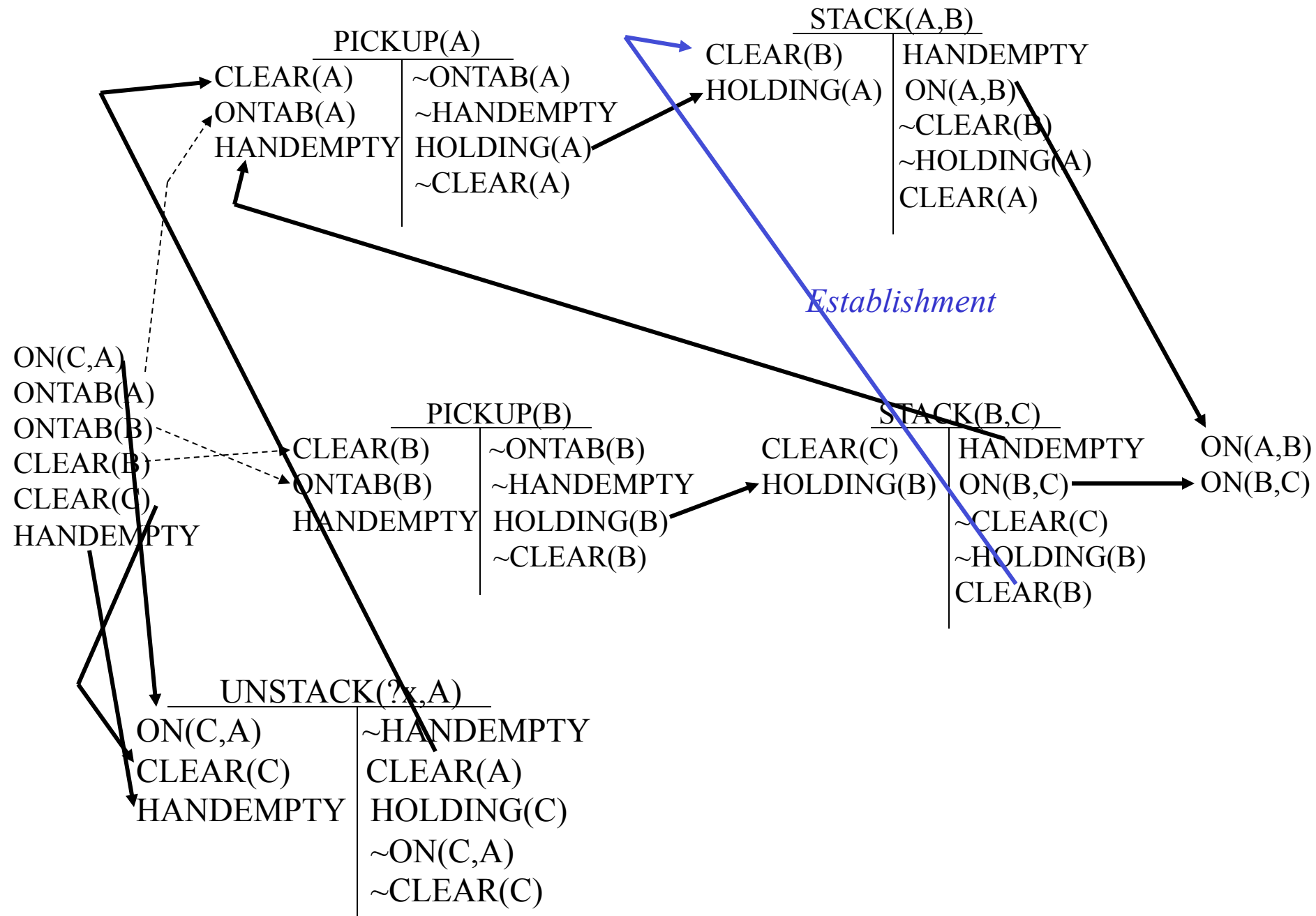


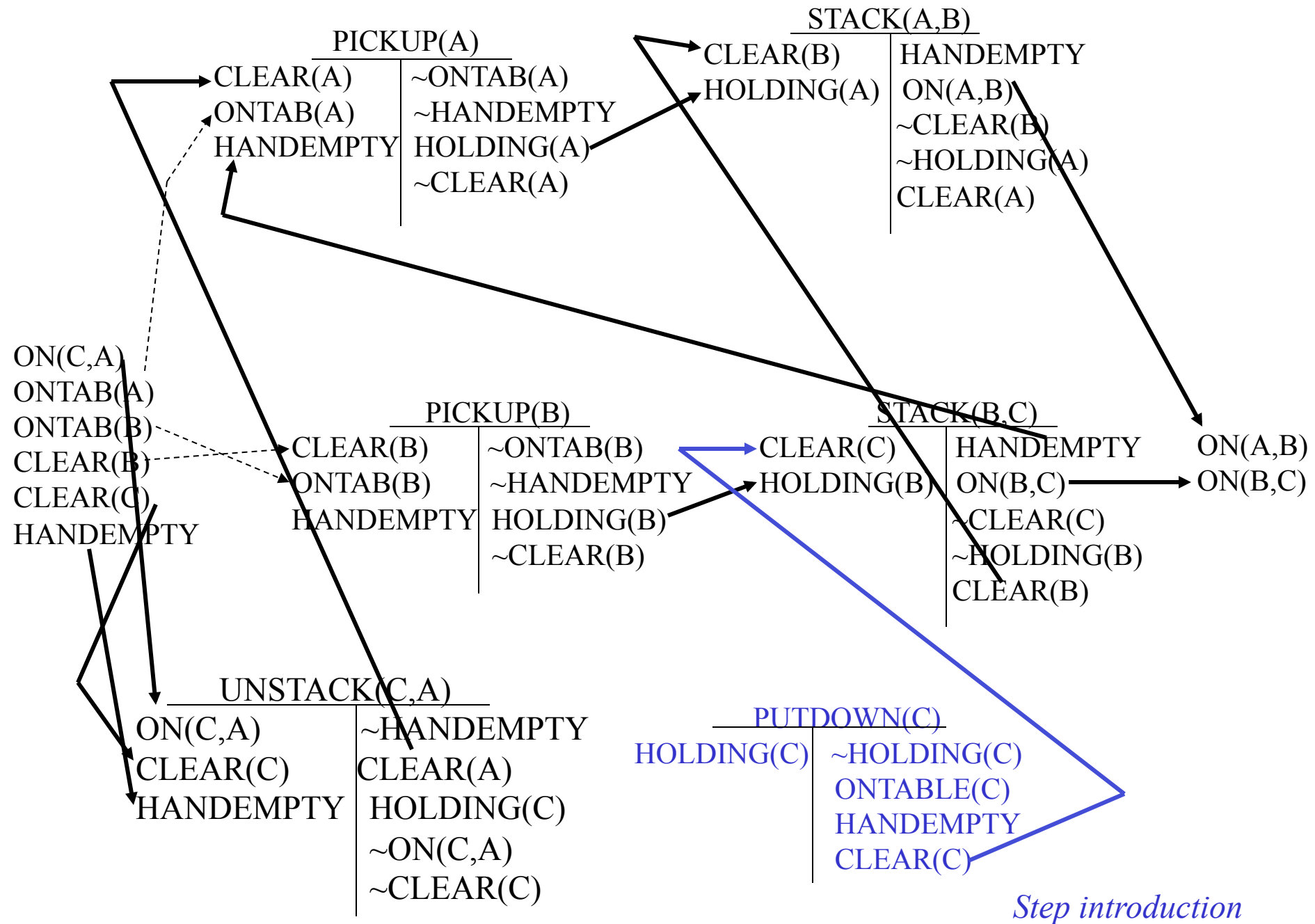
Step addition and tentative establishment

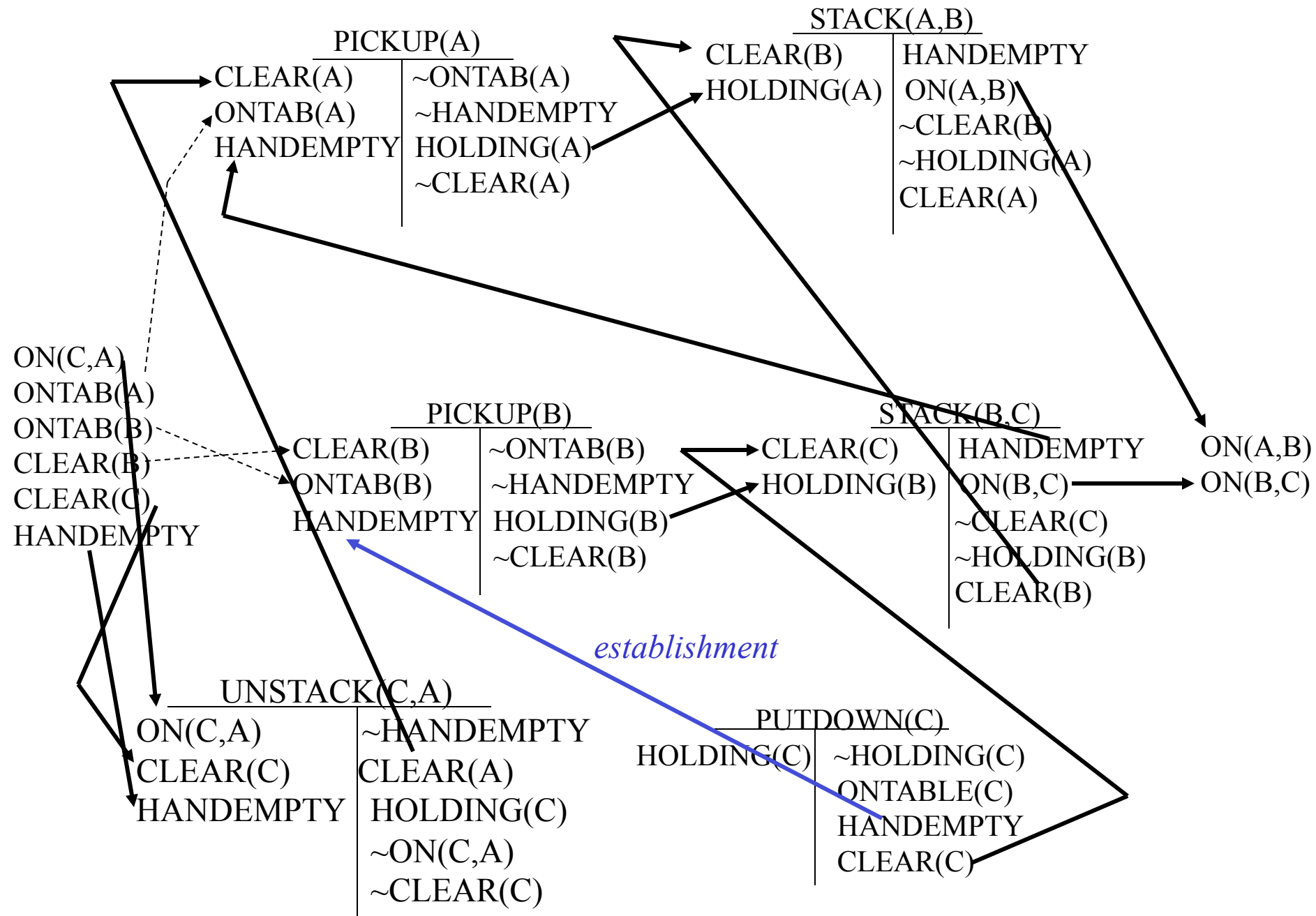


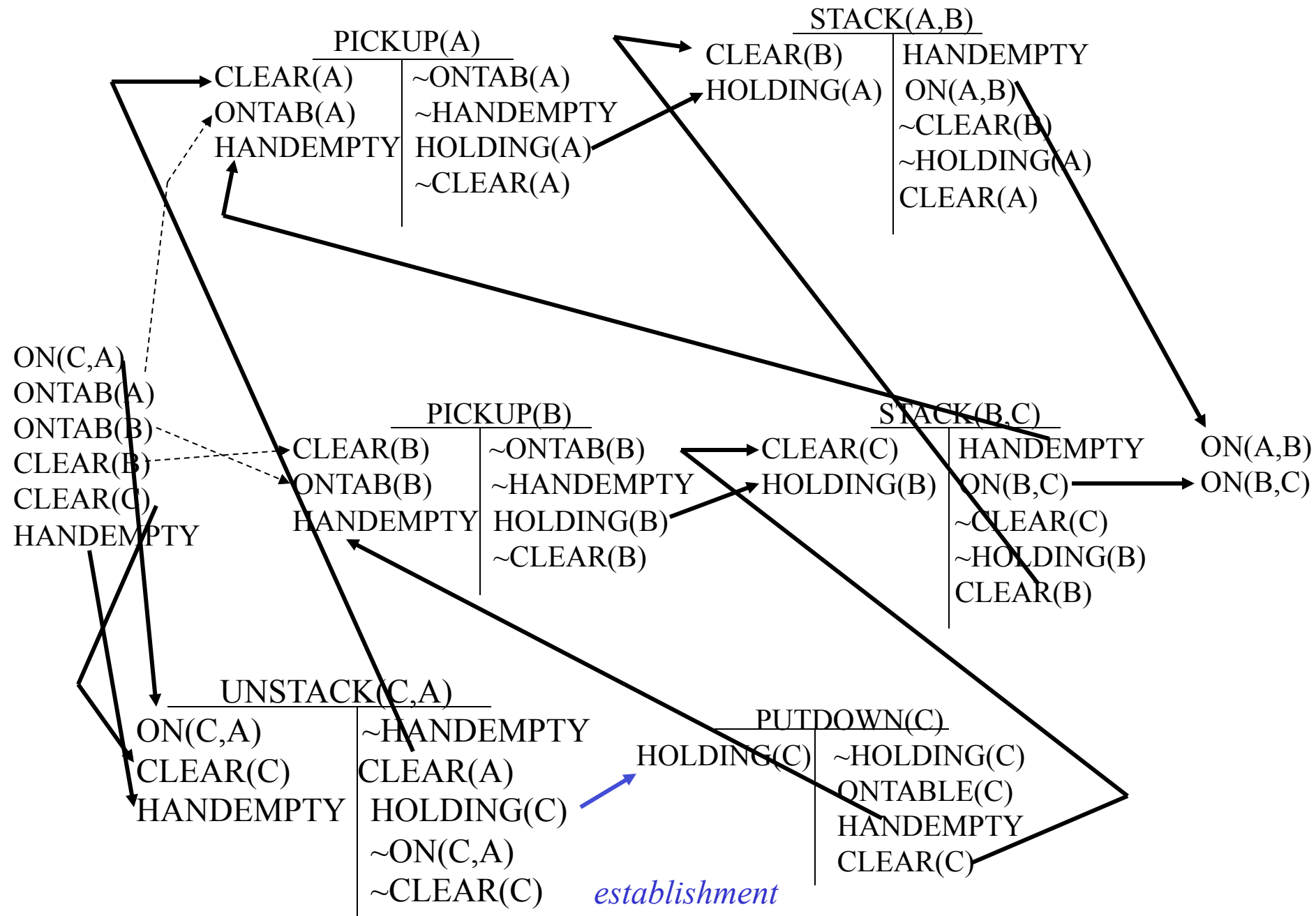




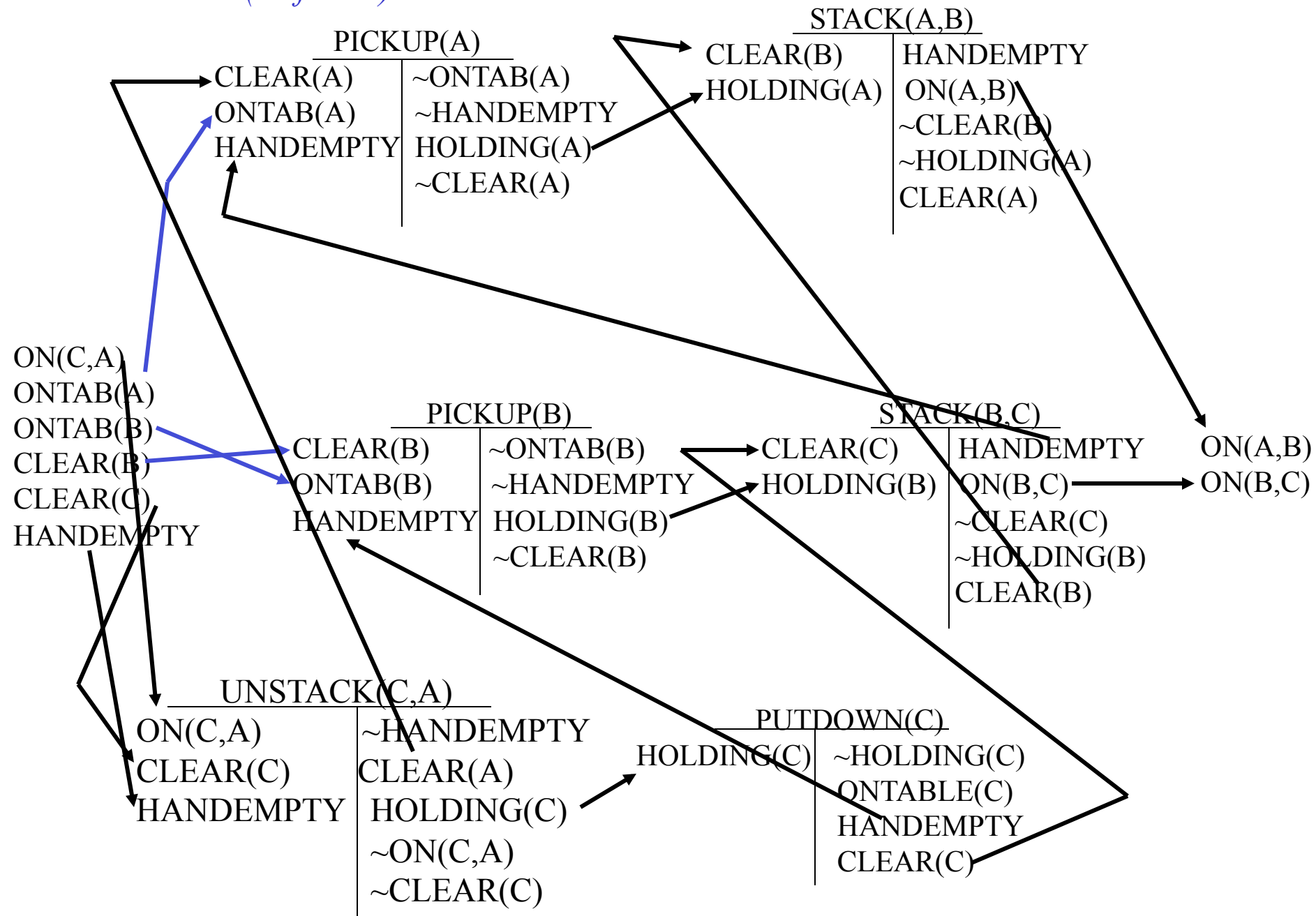






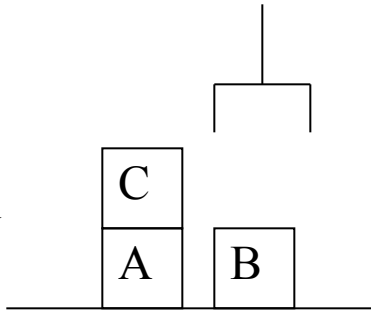


Establishment (3 of them)

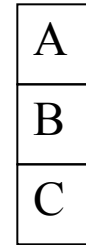


UNSTACK(C,A) → PUTDOWN(C) → PICKUP(B) → STACK(B,C) → PICKUP(A) → STACK(A,B)

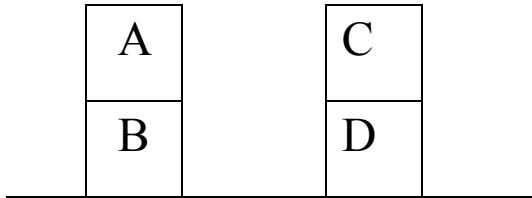
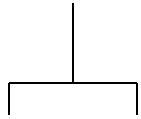
ON(C,A)
ONTAB(A)
ONTAB(B)
CLEAR(B)
CLEAR(C)
HANDEEMPTY



ON(A,B)
ON(B,C)

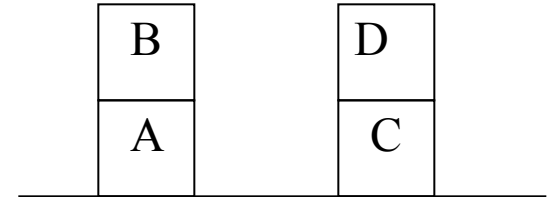


Step introduction



Initial State

ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEMPTY

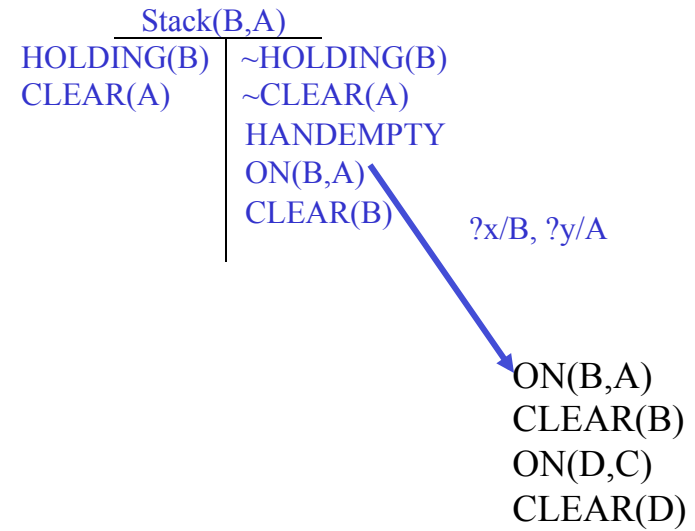


Goal spec

ON(B,A)
CLEAR(B)
ON(D,C)
CLEAR(D)

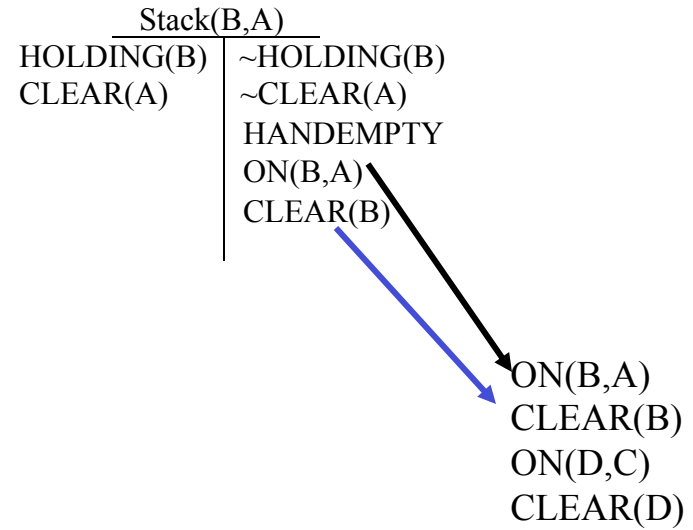
Step addition

ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEEMPTY



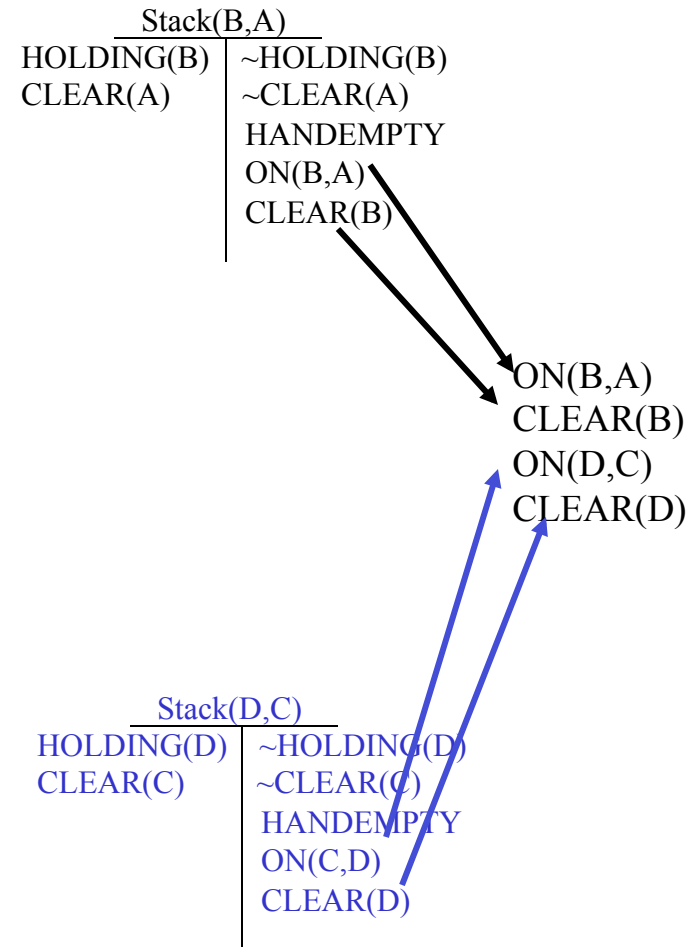
establishment

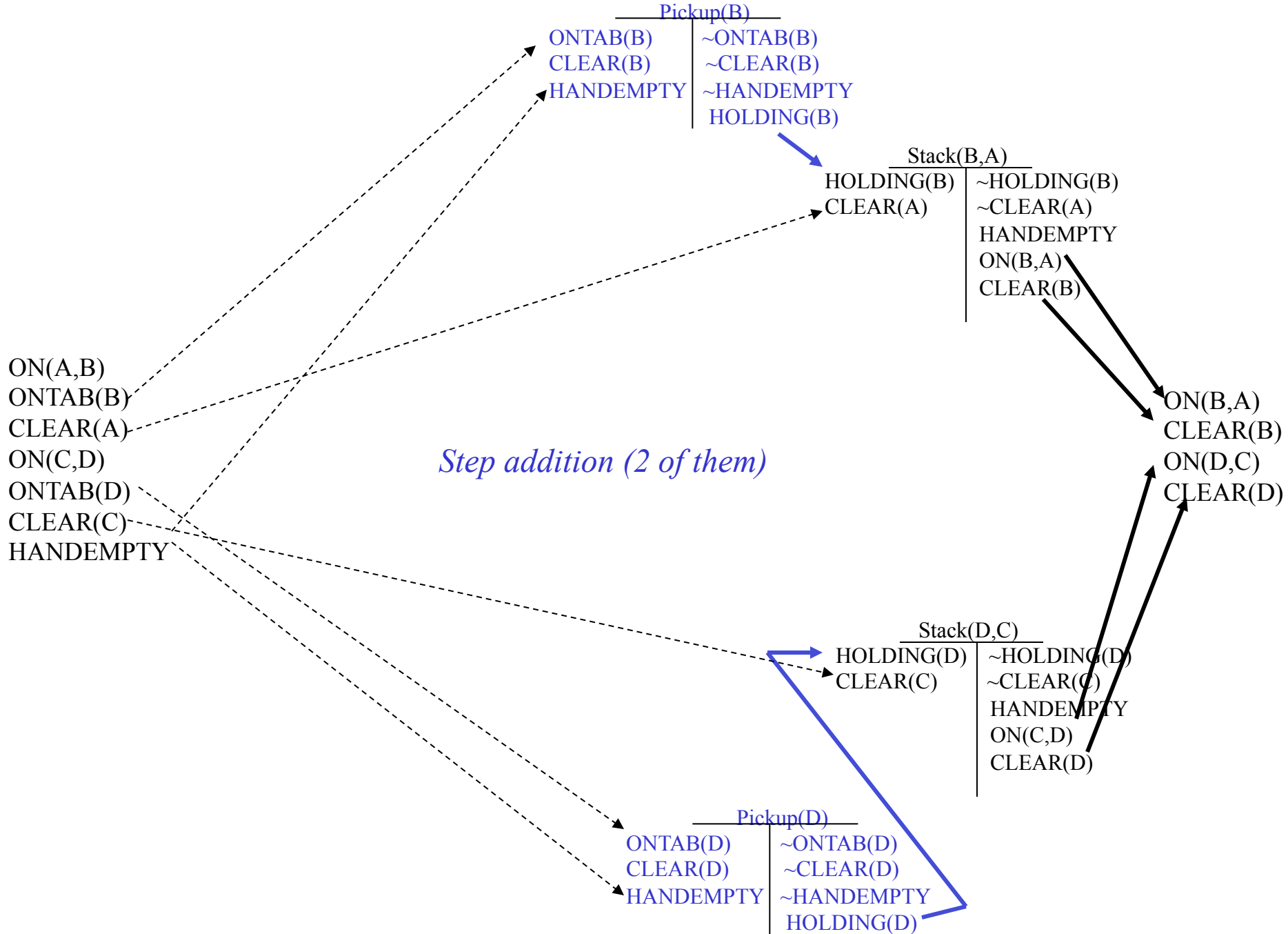
ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEEMPTY

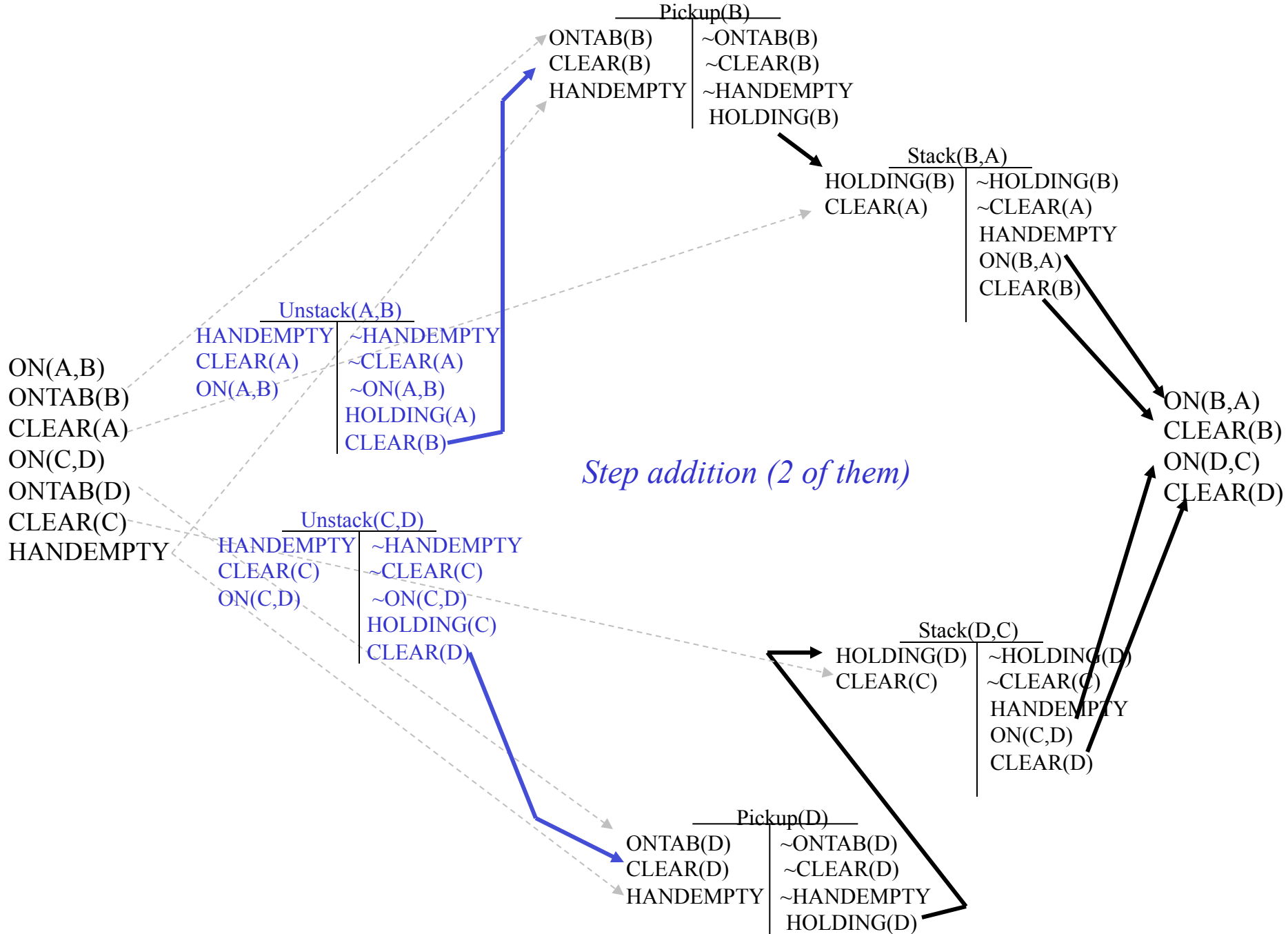


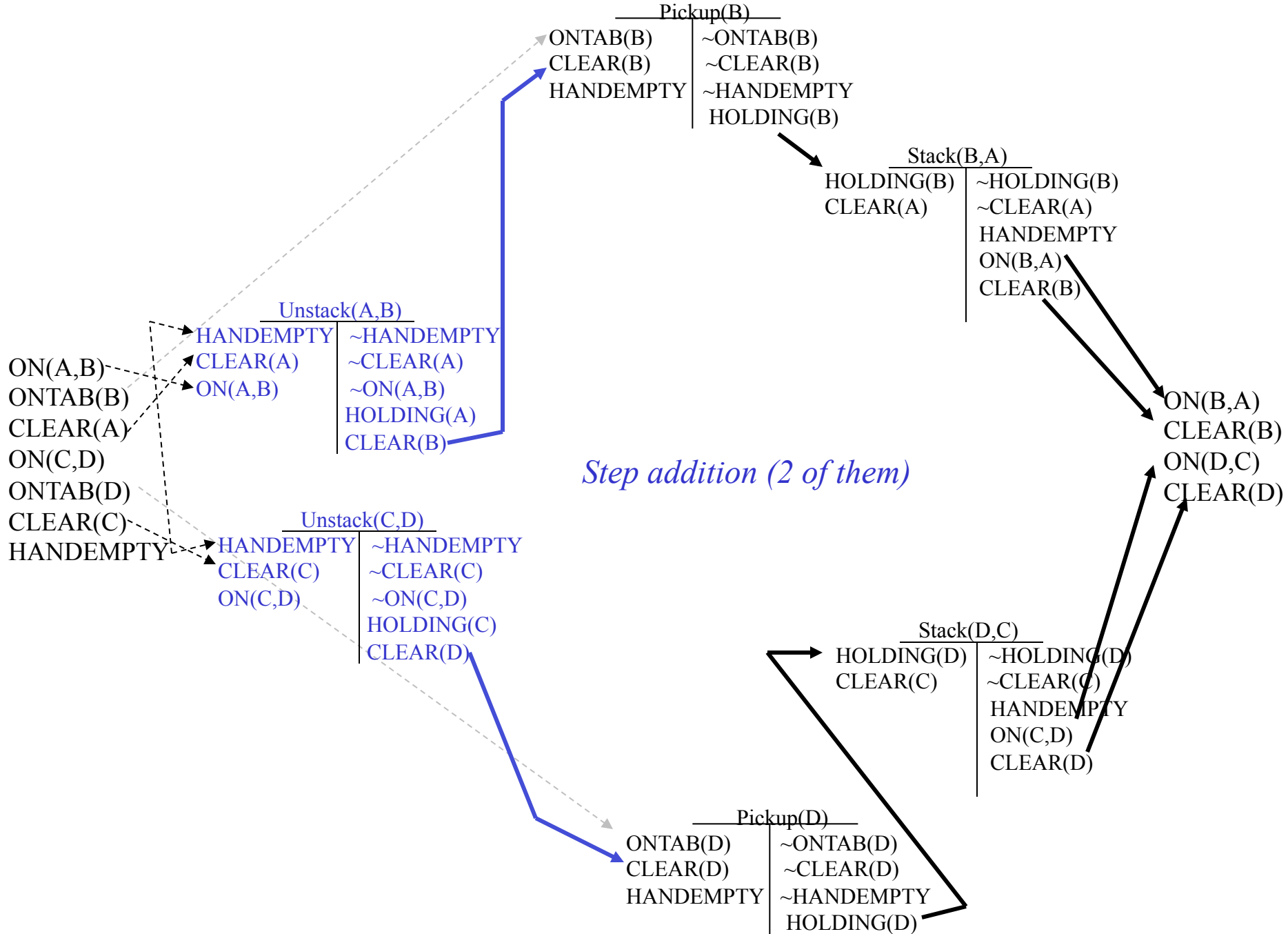
Step addition + establishment

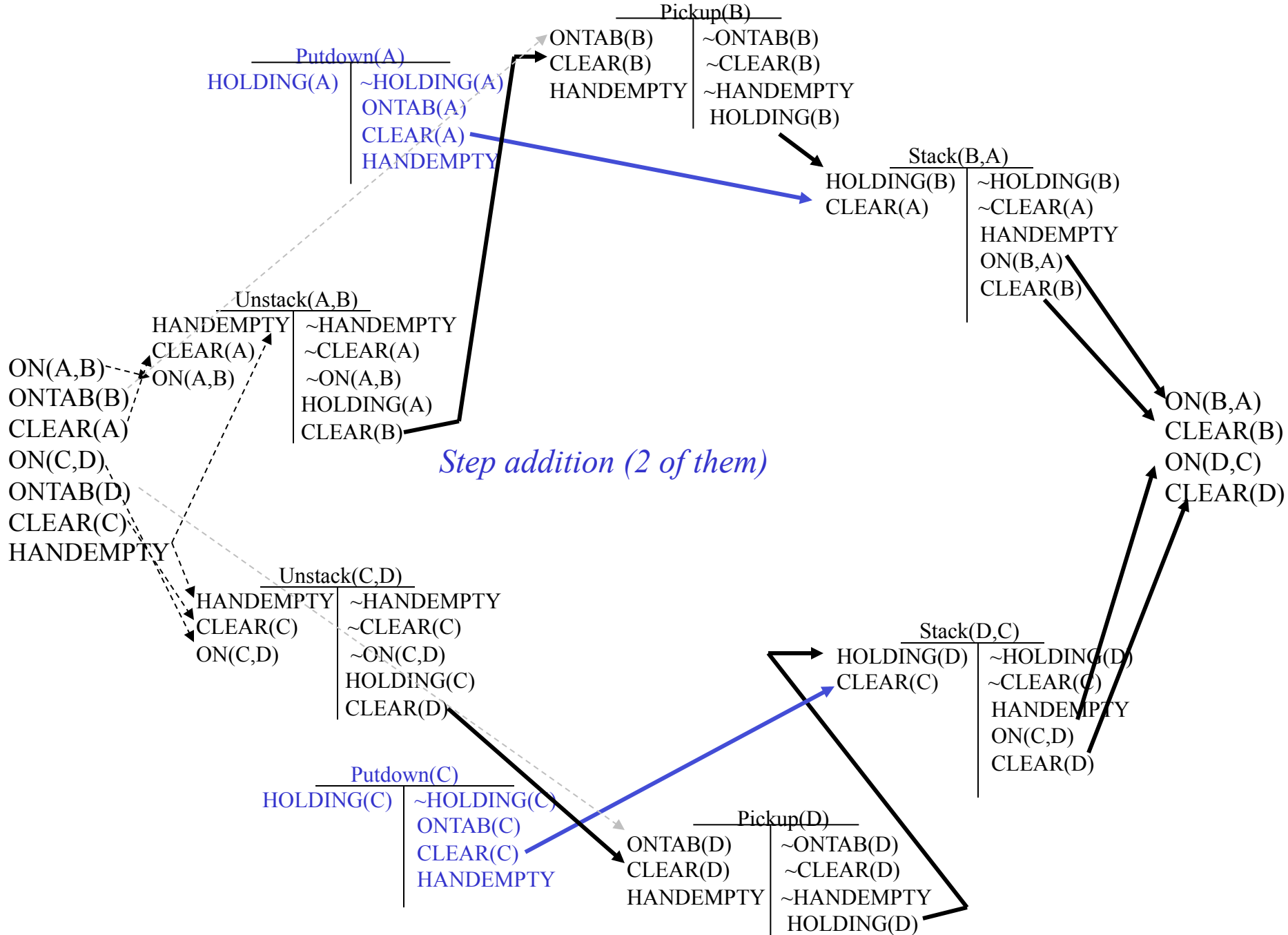
ON(A,B)
ONTAB(B)
CLEAR(A)
ON(C,D)
ONTAB(D)
CLEAR(C)
HANDEEMPTY

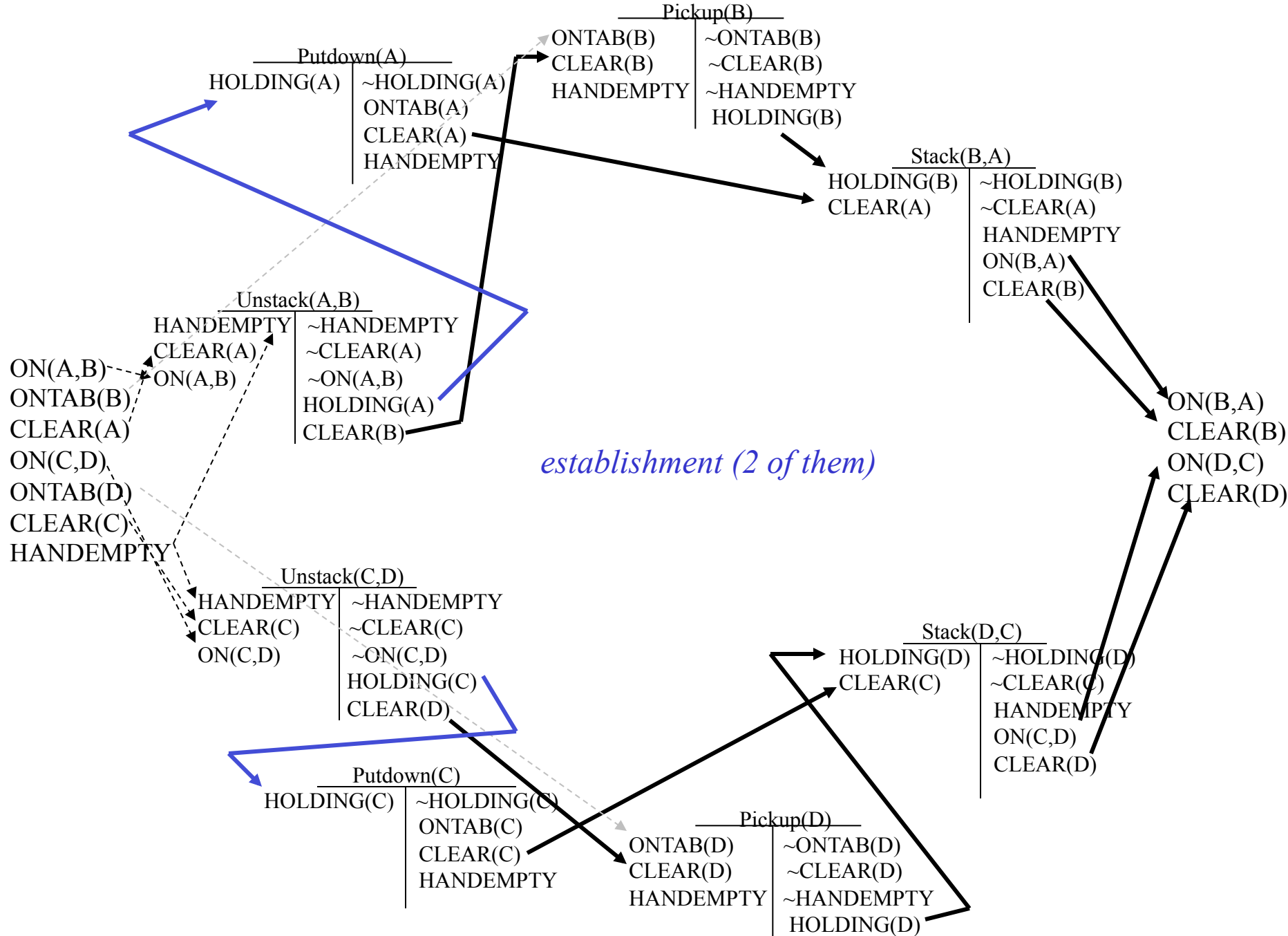


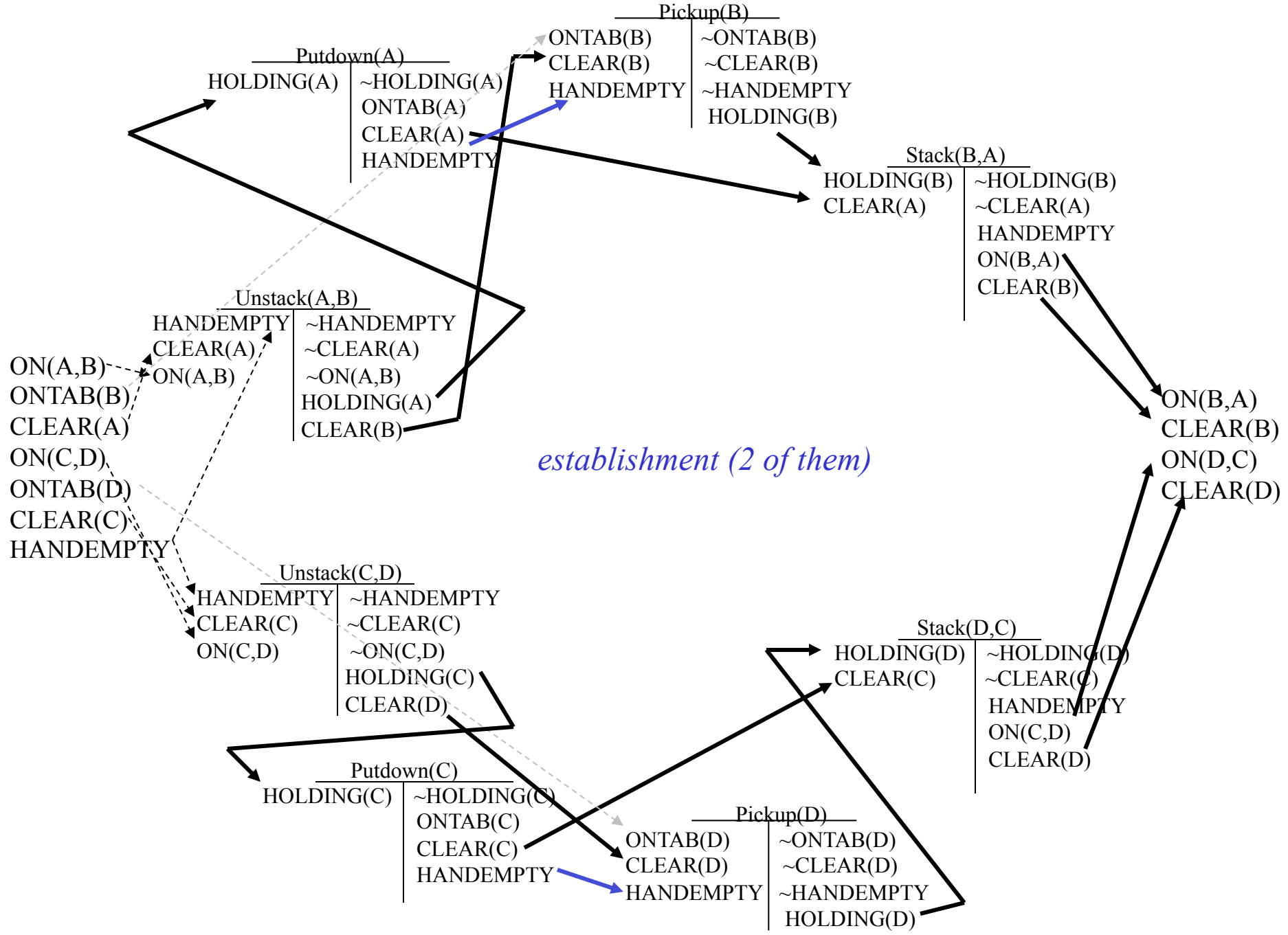


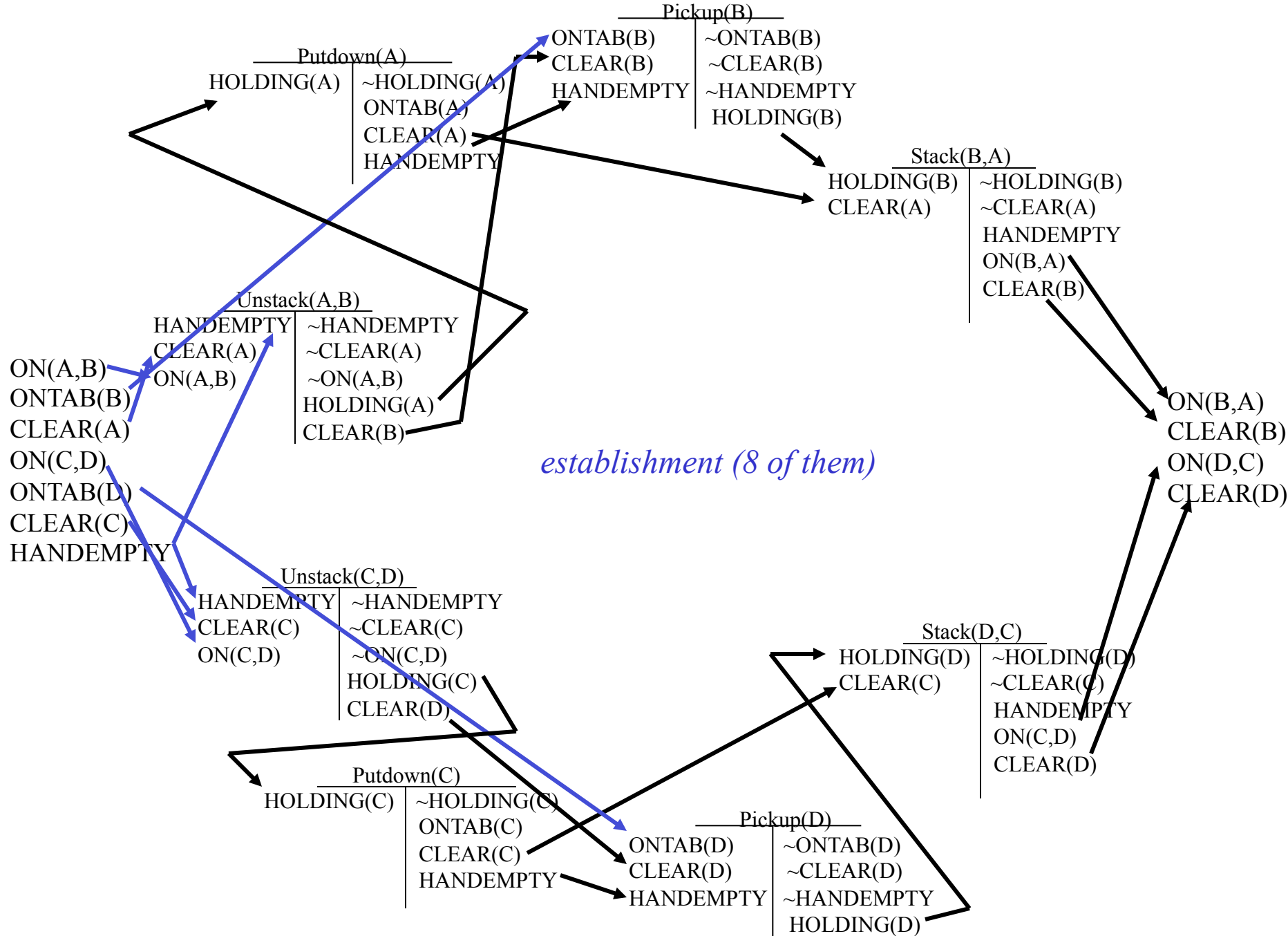








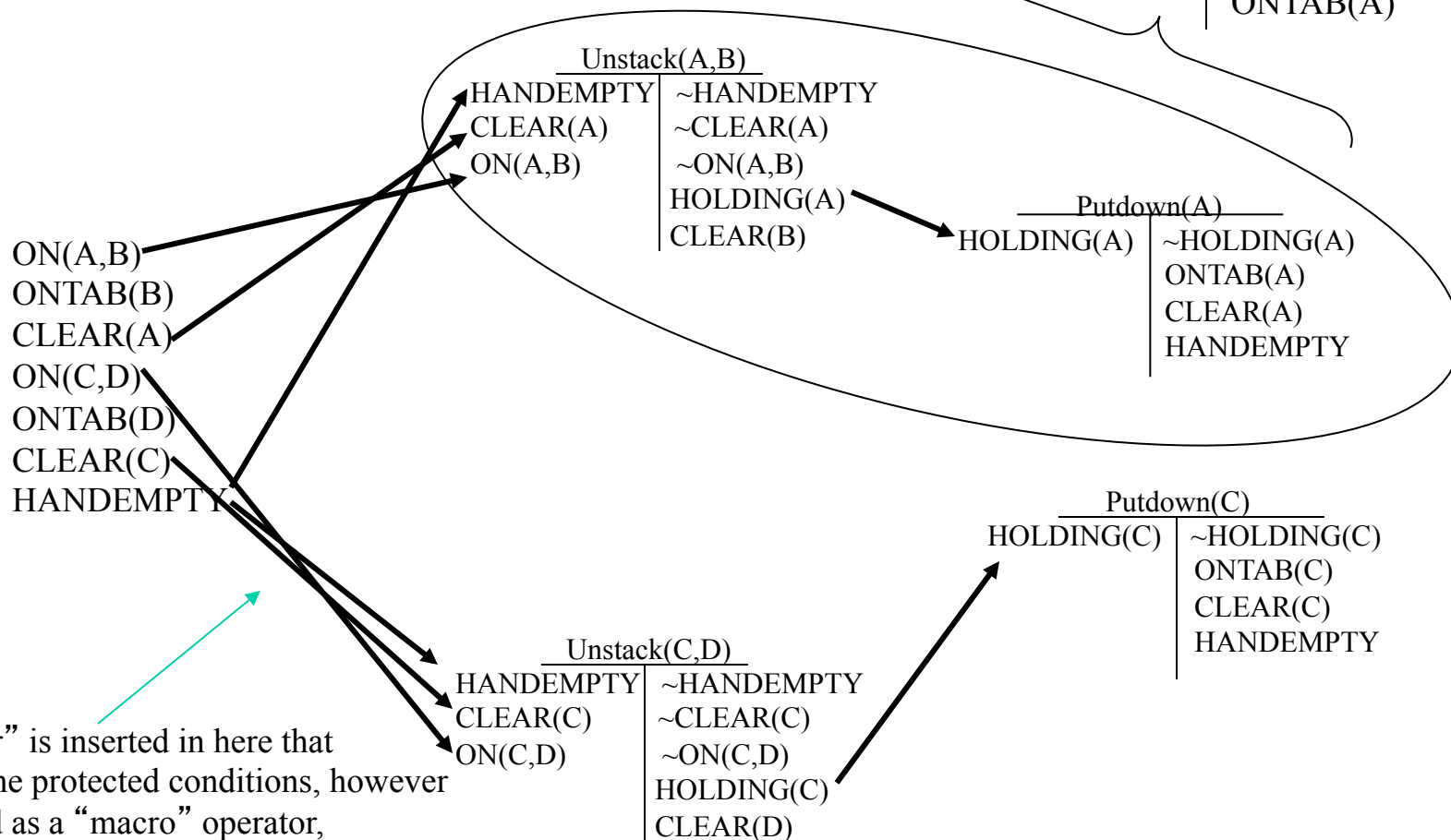




Plan execution:

A “macro” operator

<u>Unstack(A,B) → PutDown(A)</u>	
HANDEEMPTY	HANDEEMPTY
CLEAR(A)	CLEAR(A)
ON(A,B)	~ON(AB)
	CLEAR(B)
	ONTAB(A)



No “operator” is inserted in here that will negate the protected conditions, however when viewed as a “macro” operator, Unstack(A,B) → PutDown(A) does not clobber any protected subgoals (or more exactly, it restores any subgoals (e.g., handempty) that are temporally “clobbered”)

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
DEL: CONT(?r1, ?v1)
ADD: CONT(?r1, ?v2)

CONT(P,a)
CONT(Q,b)
CONT(S,c)

CONT(Q,a)
CONT(P,b)

Initial State

P

a

Q

b

S

c

Goal Spec

P

b

Q

a

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)

CONT(P,a)
 CONT(Q,b)
 CONT(S,c)

Initial State

P

a

Q

b

S

c

Assign(?r1, ?v1, ?r2, ?v2)
 CONT(?r1, ?v1) | ~CONT(?r1, ?v1)
 CONT(?r2, ?v2) | CONT(?r1, ?v2)

?r1/Q, ?v2/a

CONT(Q,a)
 CONT(P,b)

Assign(?r1, ?v1, ?r2, ?v2)
 CONT(?r1, ?v1) | ~CONT(?r1, ?v1)
 CONT(?r2, ?v2) | CONT(?r1, ?v2)

?r1/P, ?v2/b

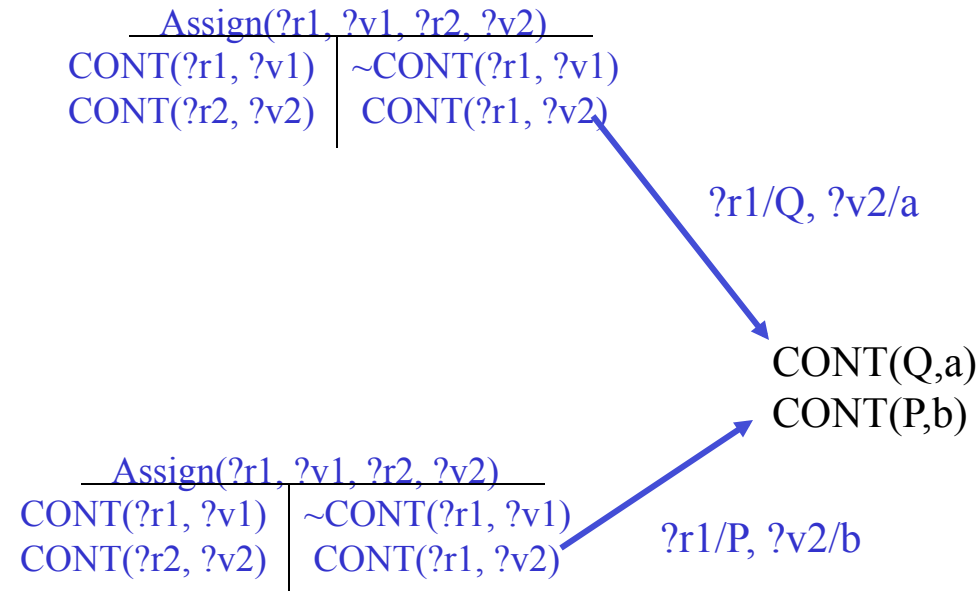
Goal Spec

P

b

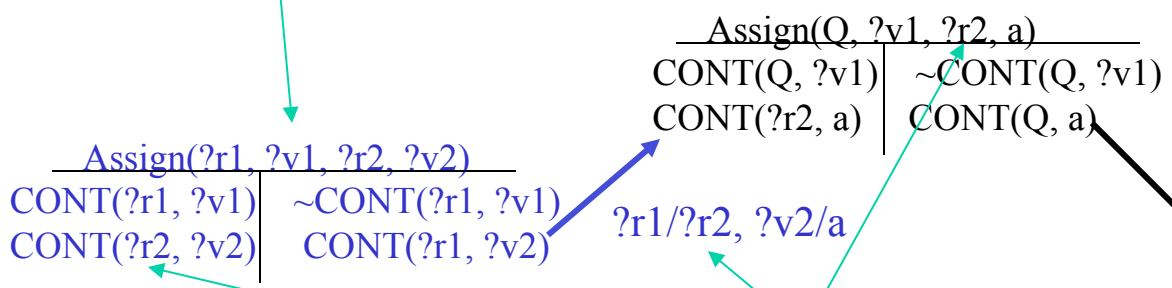
Q

a



Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)

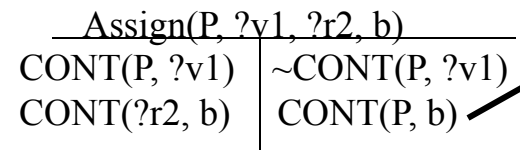
This ?v1 is different from this ?v1. Standardize apart



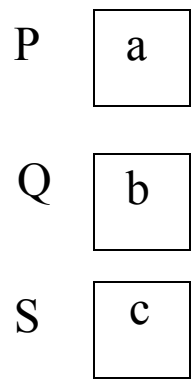
this ?r2 is different from this ?r2
Standardize apart.

CONT(P,a)
CONT(Q,b)
CONT(S,c)

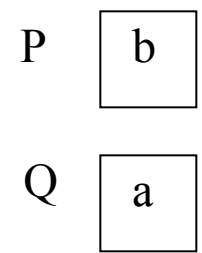
CONT(Q,a)
CONT(P,b)



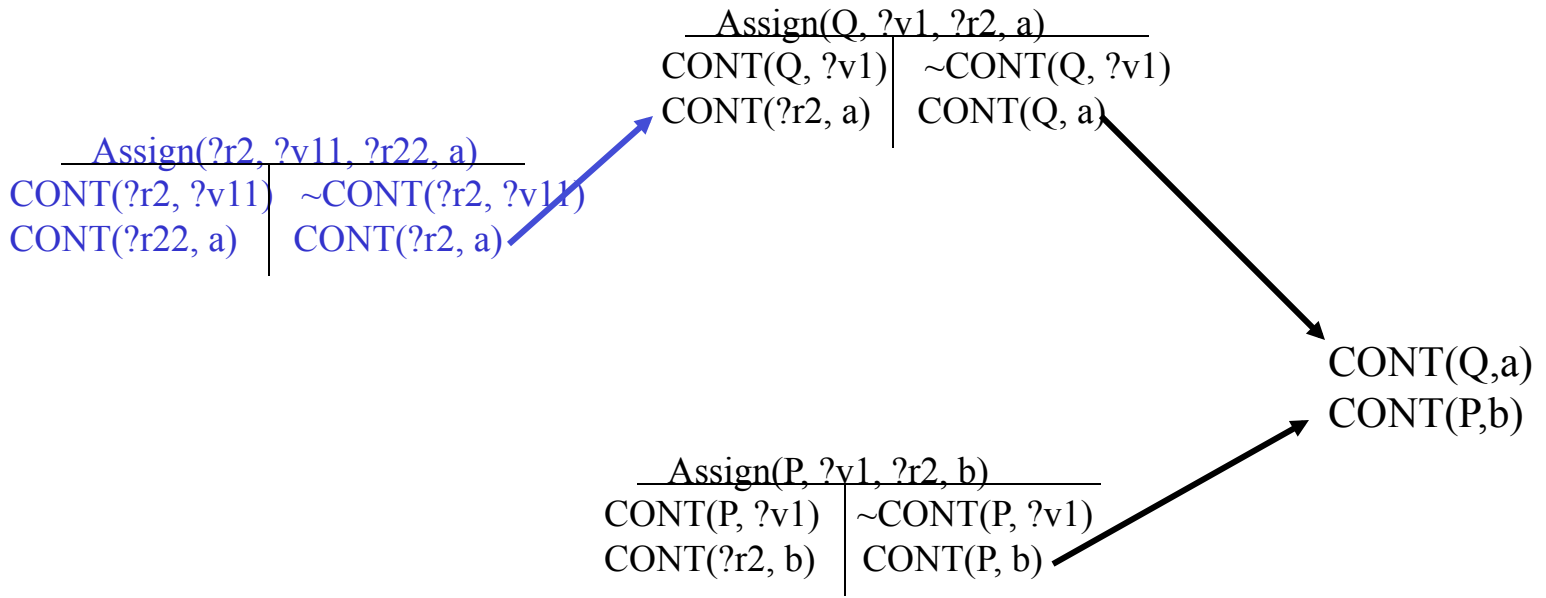
Initial State



Goal Spec



Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



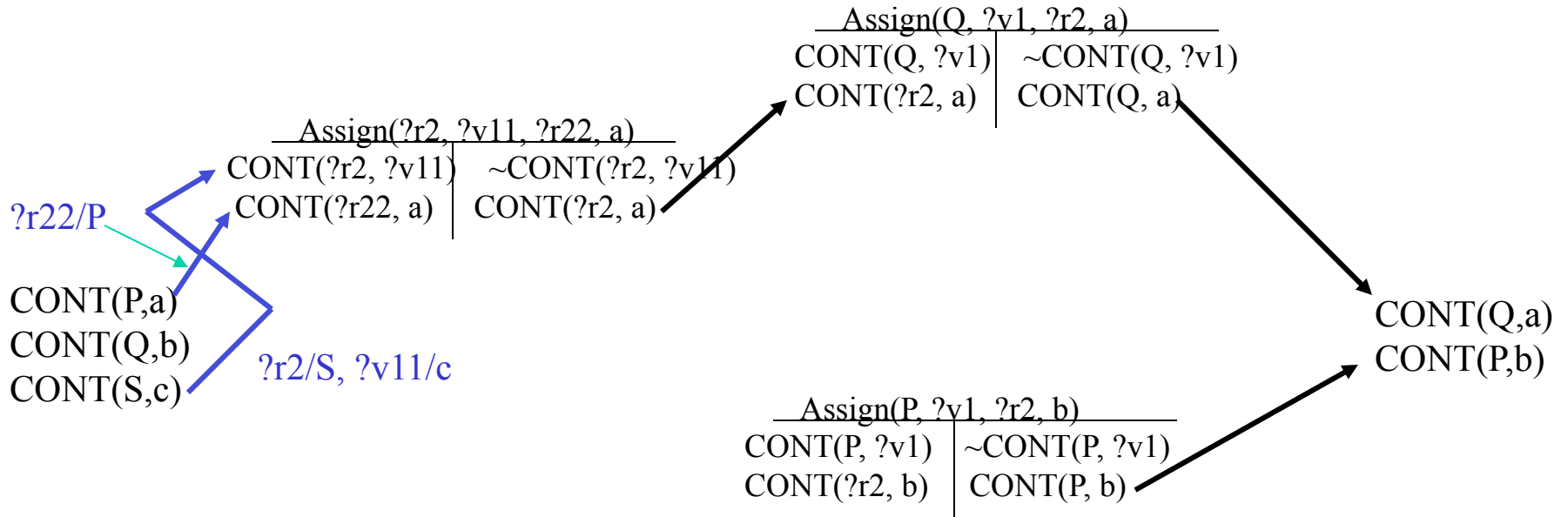
Initial State

P	a
Q	b
S	c

Goal Spec

P	b
Q	a

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



Initial State

P

a

Q

b

S

c

Goal Spec

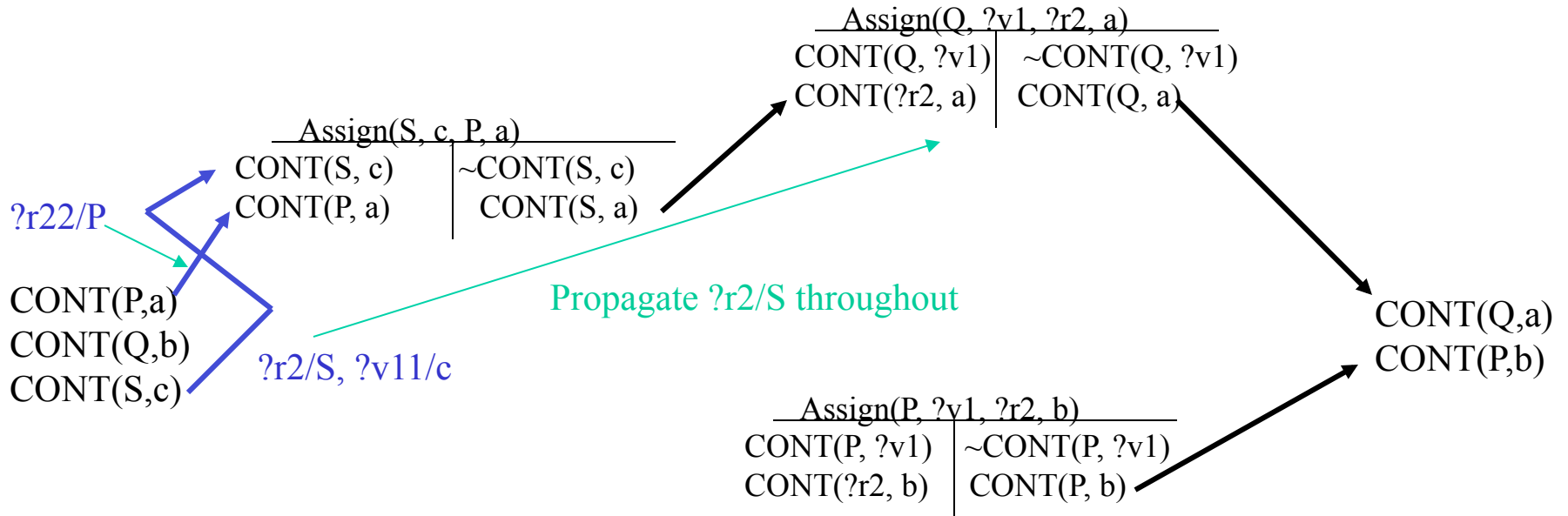
P

b

Q

a

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



Initial State

P

a

Q

b

S

c

Goal Spec

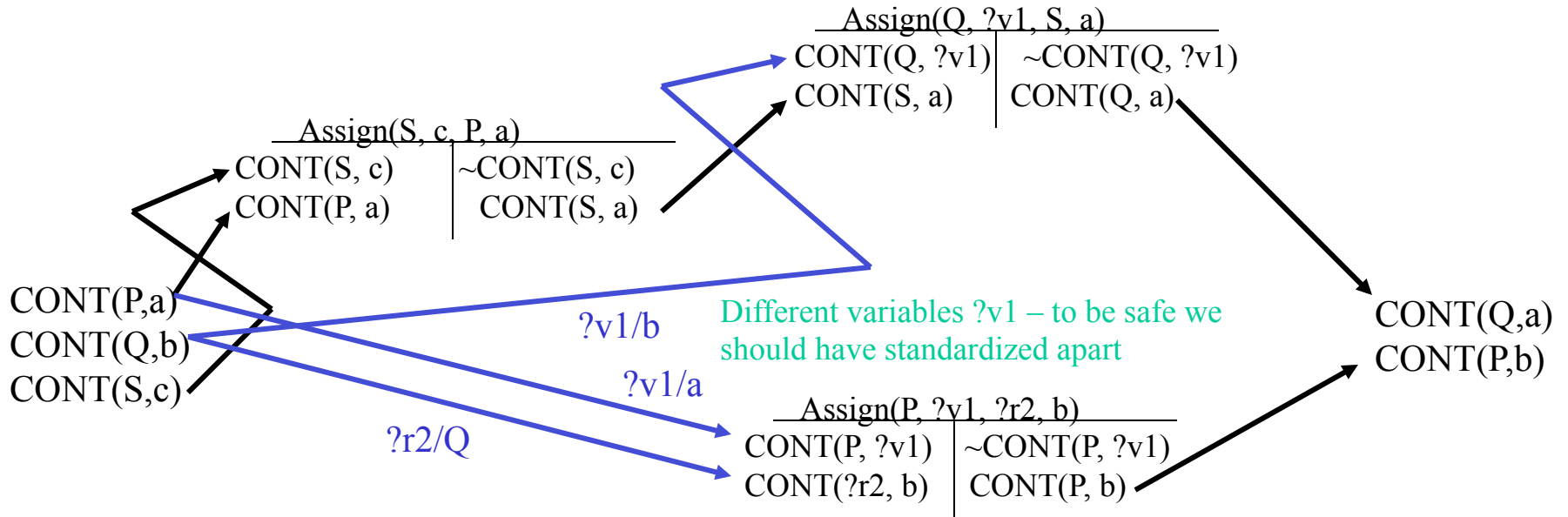
P

b

Q

a

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



Initial State

P

a

Q

b

S

c

Goal Spec

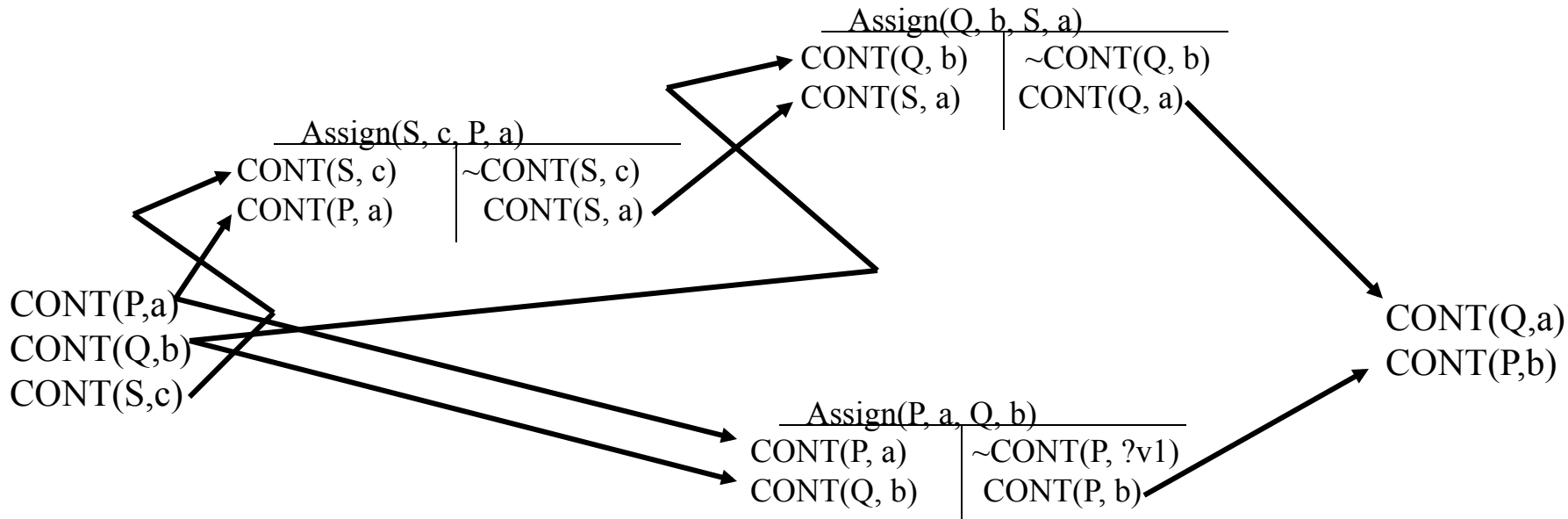
P

b

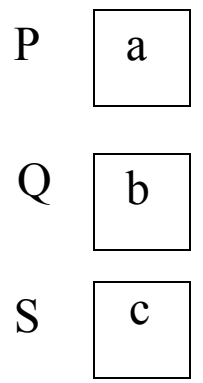
Q

a

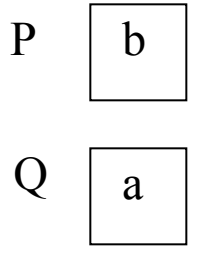
Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



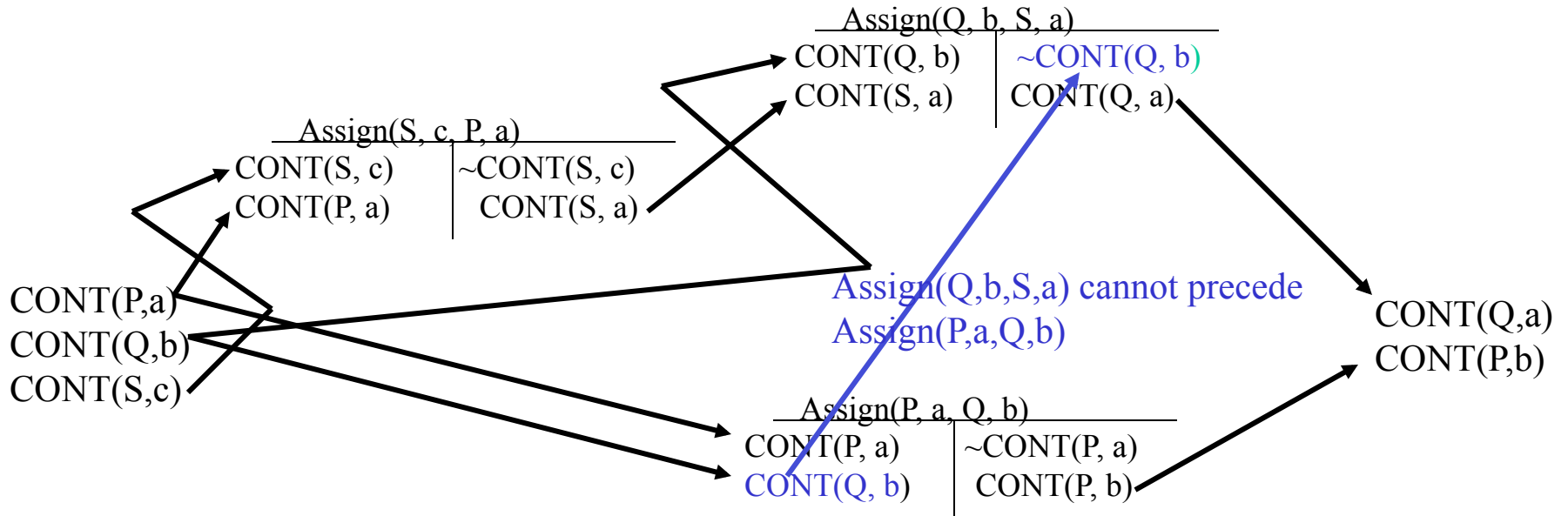
Initial State



Goal Spec



Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



Initial State

P

a

Q

b

S

c

Goal Spec

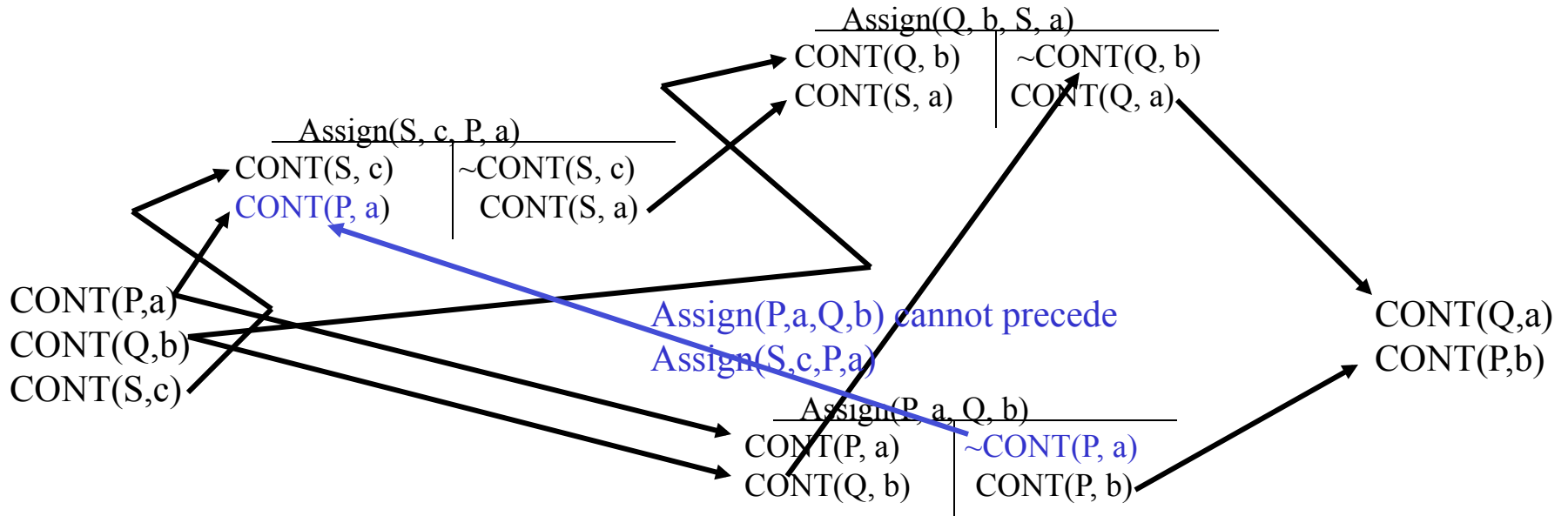
P

b

Q

a

Assign(?r1, ?v1, ?r2, ?v2) /* ?r1 = ?r2 */
 PRE: CONT(?r1, ?v1), CONT(?r2, ?v2)
 DEL: CONT(?r1, ?v1)
 ADD: CONT(?r1, ?v2)



Initial State

P	a
Q	b
S	c

Only one consistent order:

Assign(S,c,P,a) [S = P]
 → Assign(P,a,Q,b) [P = Q]
 → Assign(Q,b,S,a) [Q = S]

Goal Spec

P	b
Q	a