

# NON LINEAR PLANNING: EXERCISE

Given the following initial state **robot\_at(loc\_a)**, **handempty**, **object\_at(plate, loc\_a)**, **object\_at(cube, loc\_b)**,

We have to reach the goal: **coloured(plate, blue)**, **coloured(cube, red)**

Actions are modelled as follows:

## **colour(Object, Location, Colour)**

PRECOND: **object\_at(Object, Location)**, **robot\_at(Location)**,

**robot\_has(Colour)**

ADD: **coloured(Object, Colour)**

## **go(X,Y)**

PRECOND: **robot\_at(X)**

DELETE: **robot\_at(X)**

ADD: **robot\_at(Y)**

## **pickColour(C)**

PRECOND: **handempty**

DELETE: **handempty**

ADD: **robot\_has(C)**

## **releaseColour(C)**

PRECOND: **robot\_has(C)**

DELETE: **robot\_has(C)**

ADD: **handempty**

Solve the problem with the POP algorithm, identifying threats and their solution during the process

**start**

robot\_at(loc\_a), handempty, object\_at(plate, loc\_a), object\_at(cube, loc\_b)

L/loc\_a

handempty

**pickColour(blue)**

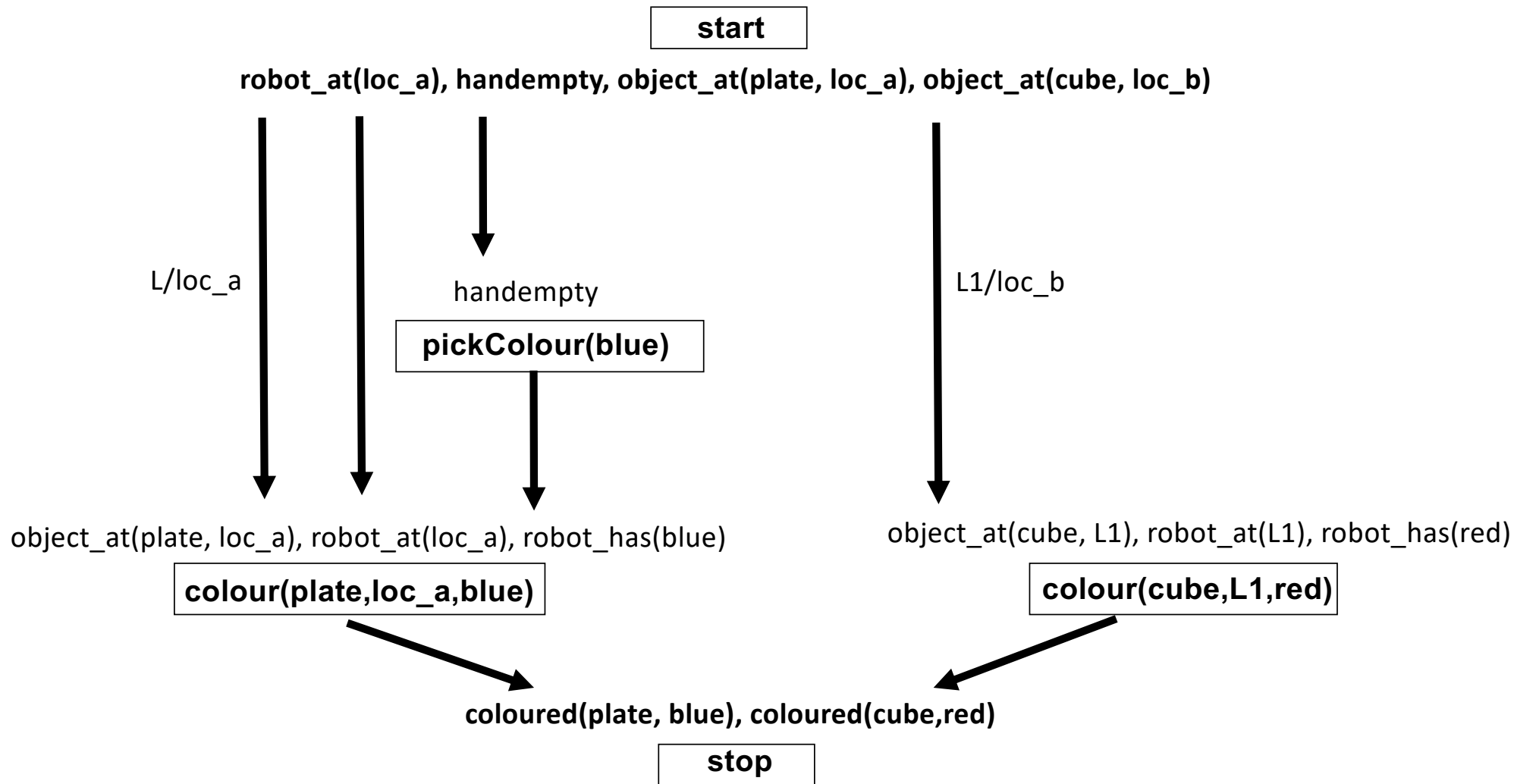
object\_at(plate, L), robot\_at(L), robot\_has(blue)

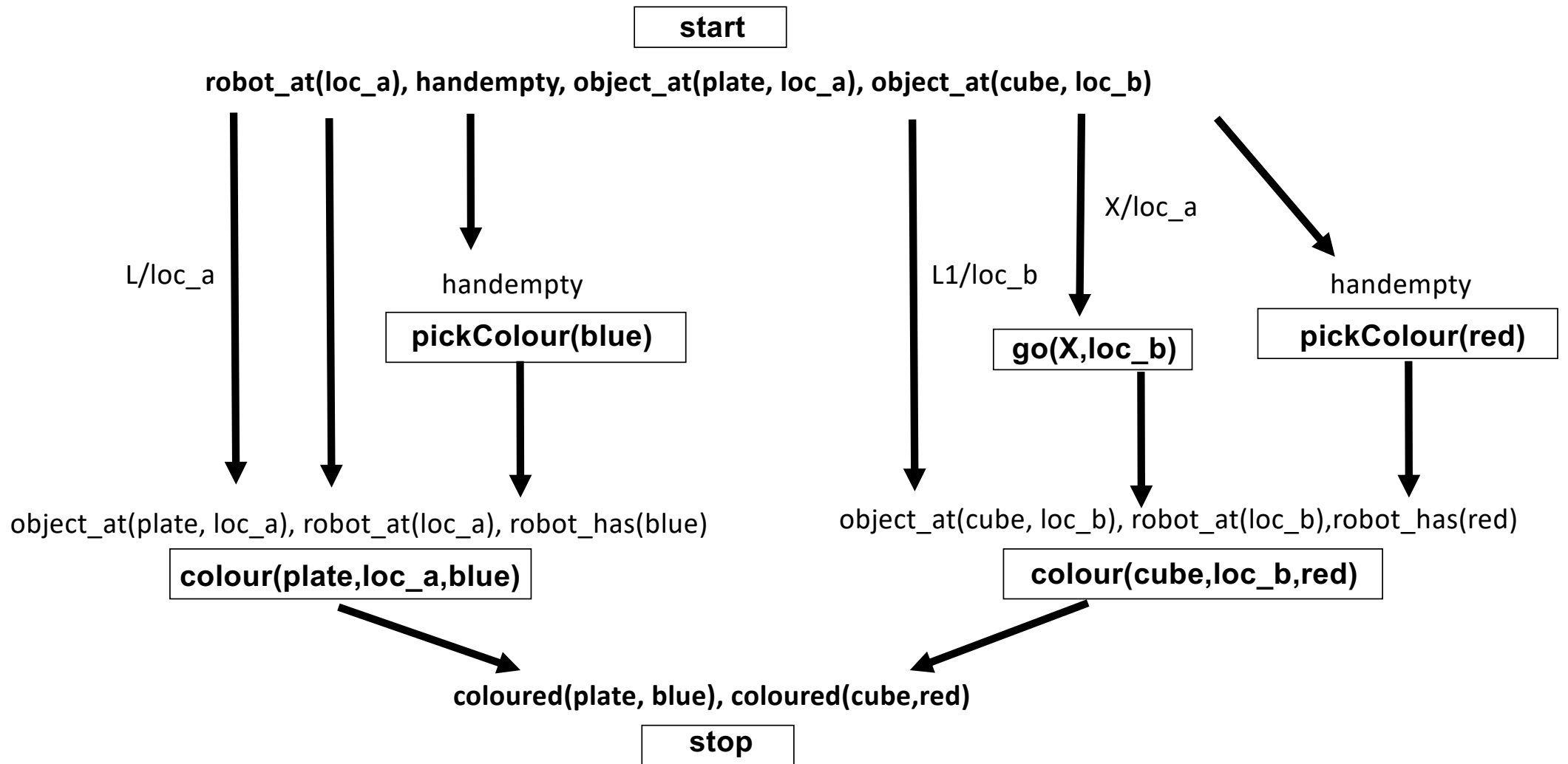
**colour(plate,L,blue)**

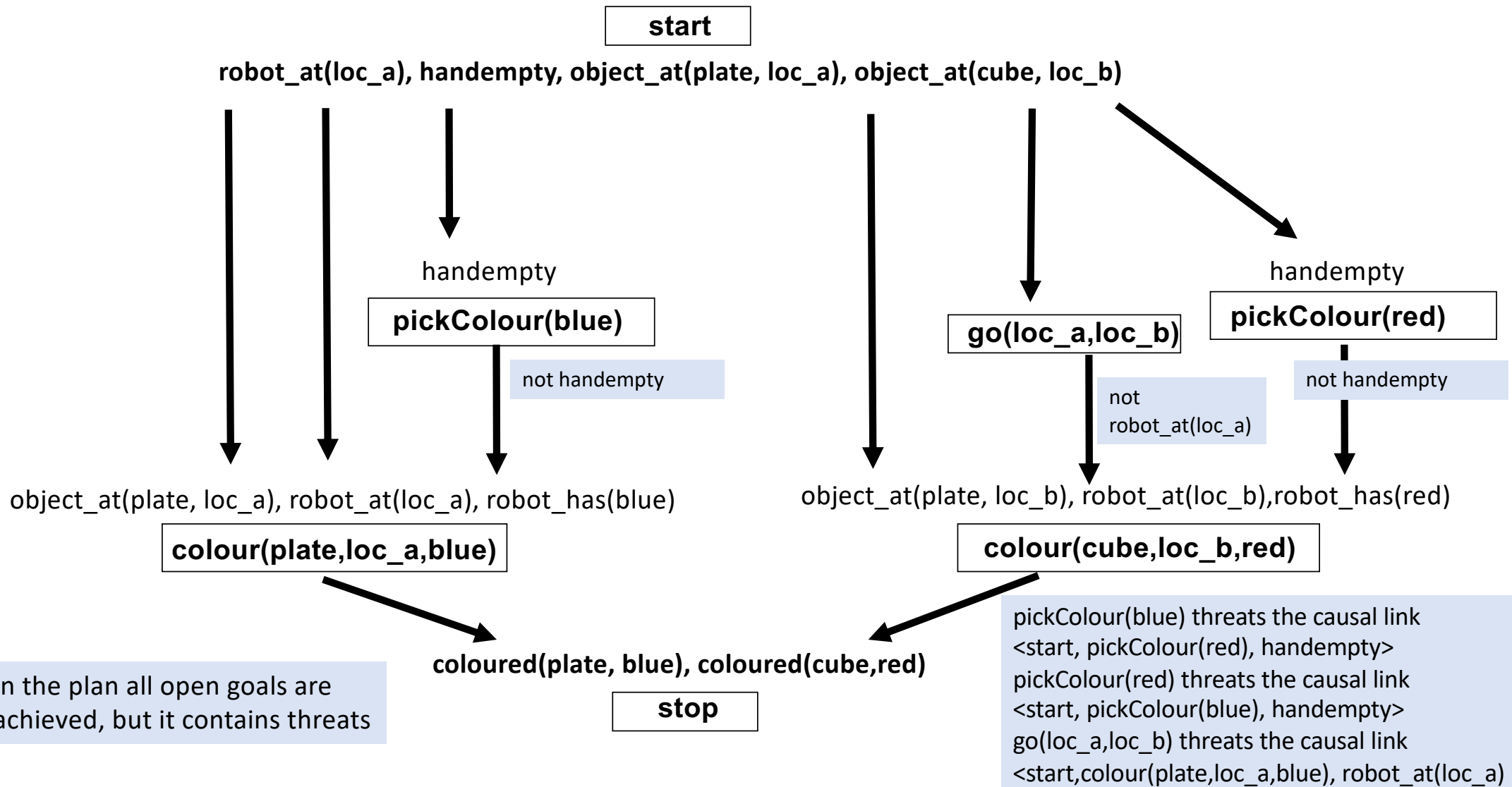
coloured(plate, blue), coloured(cube,red)

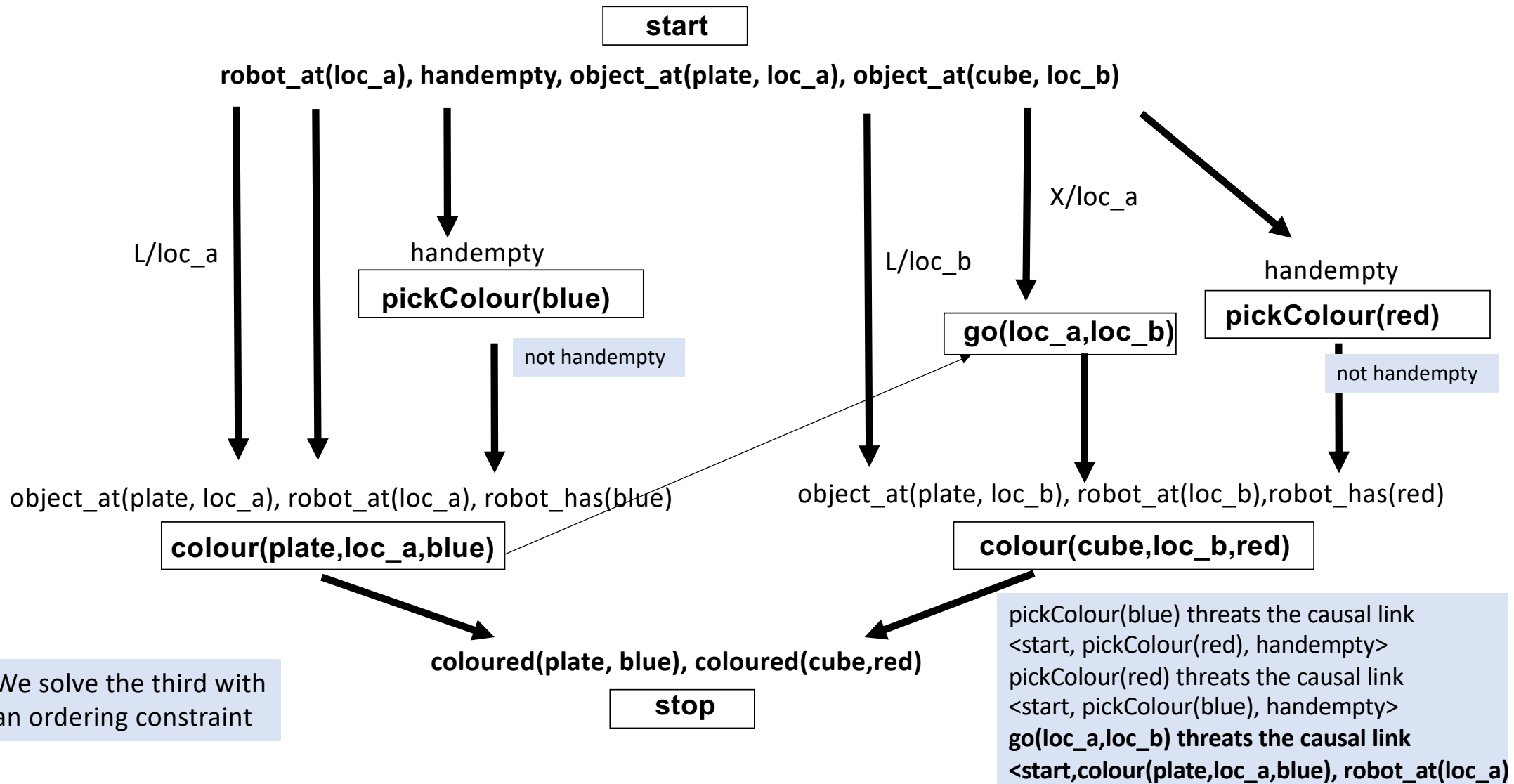
**stop**

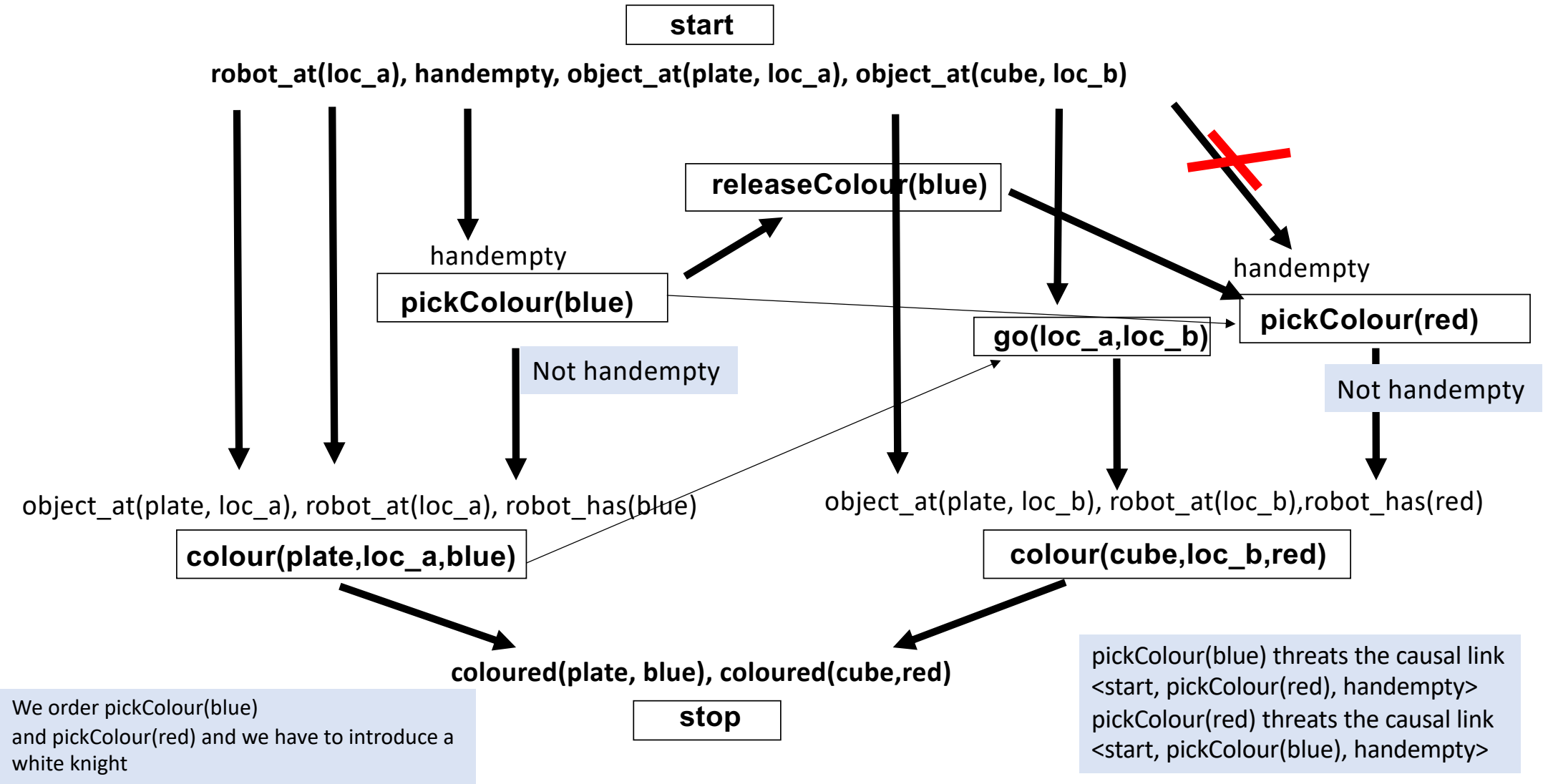


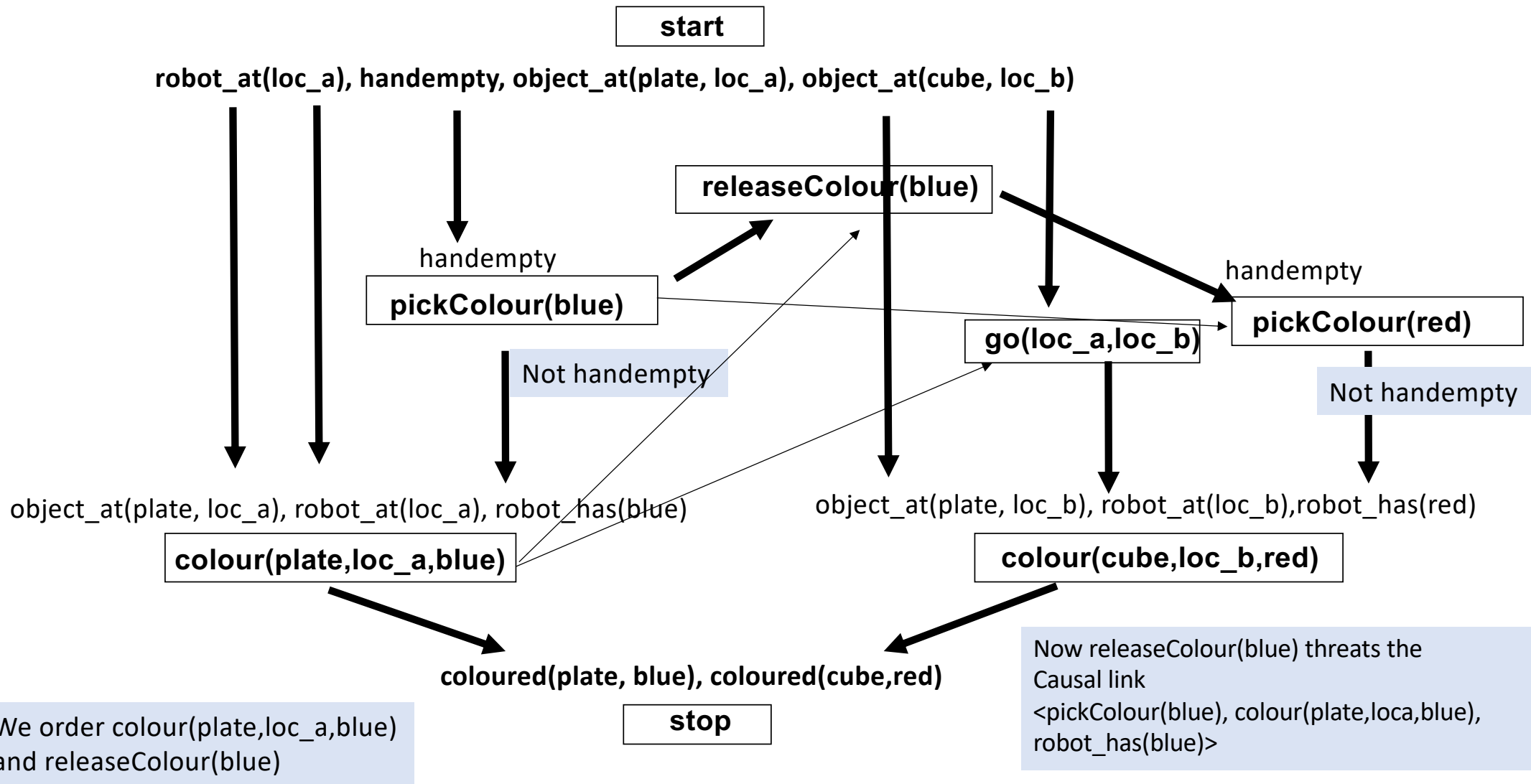






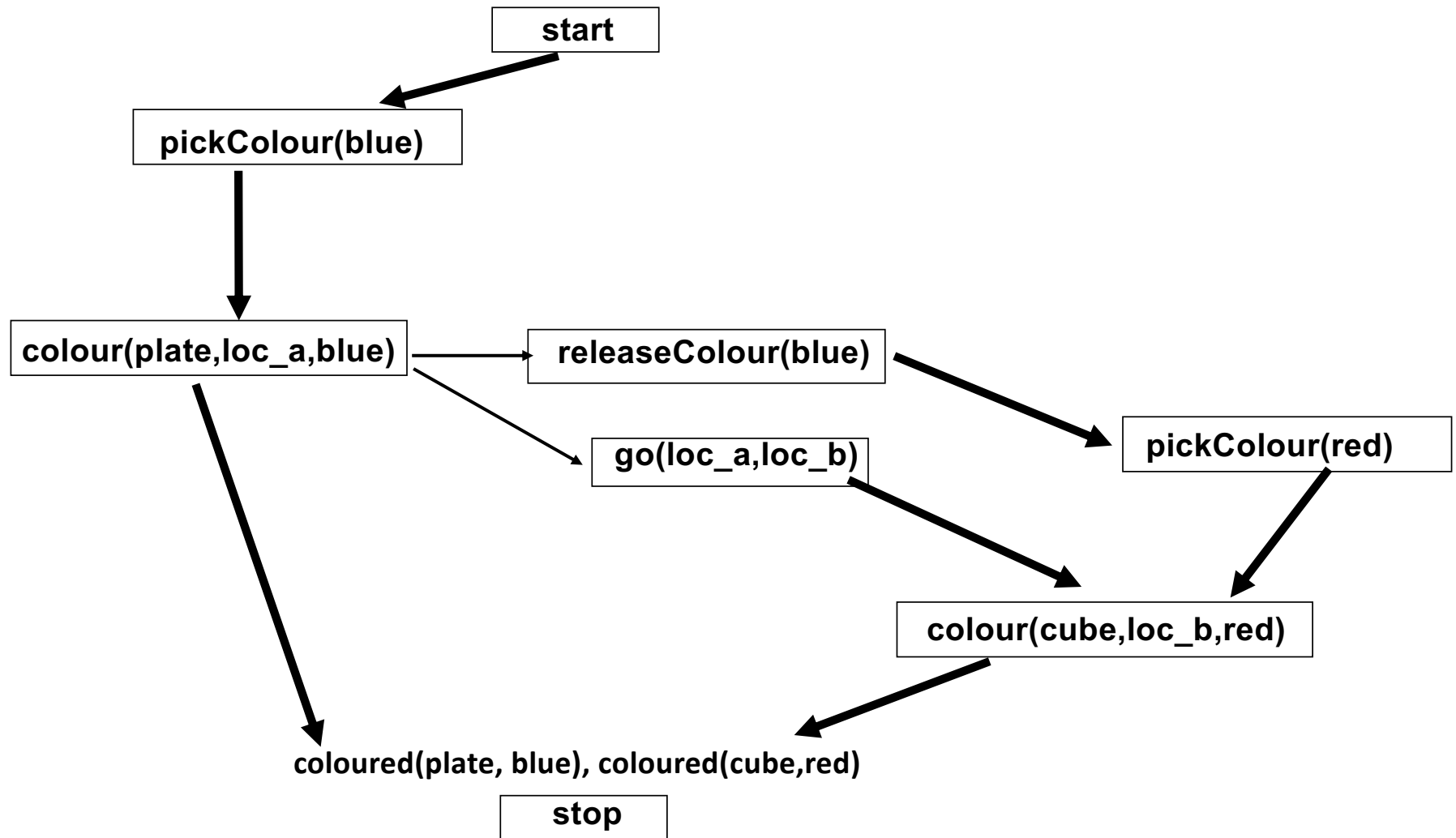








To sum up



# GRAPHPLAN: EXERCISE

Given the following initial state **robot\_at(loc\_a)**, **handempty**, **object\_at(plate, loc\_a)**, **object\_at(cube, loc\_b)**,

We have to reach the goal: **coloured(plate, blue)**, **coloured(cube, red)**

Actions are modelled as follows:

## **colour(Object, Location, Colour)**

PRECOND: **object\_at(Object, Location)**, **robot\_at(Location)**,  
**robot\_has(Colour)**  
ADD: **coloured(Object, Colour)**

## **go(X,Y)**

PRECOND: **robot\_at(X)**  
DELETE: **robot\_at(X)**  
ADD: **robot\_at(Y)**

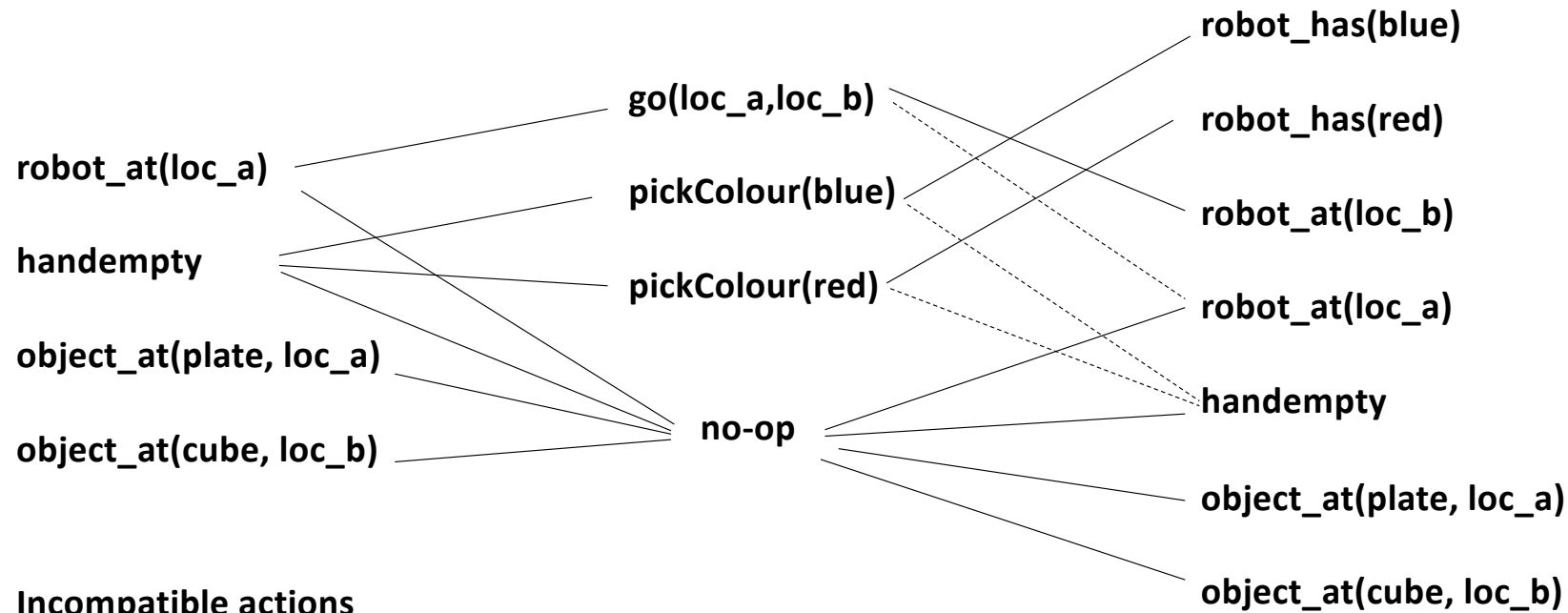
## **pickColour(C)**

PRECOND: **handempty**  
DELETE: **handempty**  
ADD: **robot\_has(C)**

## **releaseColour(C)**

PRECOND: **robot\_has(C)**  
DELETE: **robot\_has(C)**  
ADD: **handempty**

Show two levels of graphplan for this problem showing incompatible actions and incompatible propositions



### Incompatible actions

Interference:

- pickColour(blue) and pickColour(red)
- pickColour(blue) and no op on handempty
- pickColour(red) and no op on handempty
- go(loc\_a, loc\_b) and no op on robot\_at(loc\_a)

### Incompatible propositions

- robot\_at(loc\_a) and robot\_at(loc\_b)
- robot\_has(blue) and handempty
- robot\_has(red) and handempty