

# Docker Worker Robot Problem

2023320001 윤준영

## 1. Domain definition (domain-02.pddl)

정의된 도메인을 살펴보면 다음과 같다.

### 1.1) Types

location: harbor에 있는 서로 연결된 장소.

pile: location에 붙어있으며, 가장 아래에 pallet를 포함하고 있음. container를 쌓을 수 있음.

robot: 최대 1개의 container를 잡을 수 있음. 한 location 당 1개의 robot만 위치할 수 있음.

crane: location에 위치하여 container를 들어올릴 수 있음.

### 1.2) Constants

pallet: container로 취급.

### 1.3) Predicates

adjacent ?l1 ?l2: location l1과 l2는 서로 인접해 있다.

attached ?p ?l: pile p는 location l에 붙어있다.

belong ?k ?l: crane k는 l에 속해있다.

atl ?r ?l: robot r은 location l에 위치해 있다.

free ?l: location l에 로봇이 위치해 있지 않다.

loaded ?r ?c: robot r이 container c를 잡고 있다.

unloaded ?r: robot r은 container를 잡고 있지 않다.

holding ?k ?c: crane k가 container c를 들고 있다.

empty ?k: crane k는 container를 들고 있지 않다.

in ?c ?p: container c는 pile p에 쌓여 있다.

top ?c ?p: pile p 가장 위에 container c가 쌓여 있다.

on ?c1 ?c2: container c1은 container c2 바로 위에 있다.

### 1.4) Operations

move ?r ?l1 ?l2: robot r을 location l1에서 l2로 이동시킴.

precondition: and( adjacent ?l1 ?l2) (atl ?r ?l1) (free ?l2) )

l1과 l2는 인접하여야 하며, r은 l1에 위치하여야 하고 l2에는 robot이 없어야 함.

effect: and( atl ?r ?l2) (free ?l1) (not (free ?l2)) (not (atl ?r ?l1)) )

r은 l2에 위치하게 되며, l1은 robot이 없게 되고, l2는 robot이 없지 않고,

r은 l1에 위치하지 않음

load ?k ?l ?c ?r: location l의 crane k가 들고 있는 container c를 robot r에 적재함.

precondition: and( (and (atl ?r ?l) (belong ?k ?l) (holding ?k ?c) (unloaded ?r) )

r은 l에 위치하여야 하며, k는 l에 속해있어야 함. k는 c를 들고 있으며,

r은 적재되지 않은 상태여야 함.

effect: and( (loaded ?r ?c) (not (unloaded ?r)) (empty ?k) (not (holding ?k ?c)) )

r은 c를 적재하며, r은 적재되지 않은 상태가 아니게 되고,

k는 container를 들고 있지 않은 상태가 되며, k는 c를 들고 있지 않게 됨.

unload ?k ?l ?c ?r: location l의 crane k가 robot r이 적재하고 있는 container c를 들음.

precondition: and( (belong ?k ?l) (atl ?r ?l) (loaded ?r ?c) (empty ?k) )

k는 l에 속해있어야 하며, r은 l에 있어야 함. r은 c를 적재하고 있으며,

k는 container를 들고있지 않아야 함.

effect: and( (unloaded ?r) (holding ?k ?c) (not (loaded ?r ?c)) (not (empty ?k)) )

r은 적재되지 않은 상태가 되고, k는 c를 들고 있게 됨. r은 c를 적재하지 않은

상태가 되고, k는 container를 들고있지 않은 상태가 아니게 됨.

take ?k ?l ?c1 ?c2 ?p: location l의 crane k가 pile p위의 container c2위의 c1을 들음.

precondition: and( (belong ?k ?l) (attached ?p ?l) (empty ?k) (in ?c1 ?p) (top ?c1 ?p) (on ?c1 ?c2) )

k는 l에 속해있어야 하며, p는 l에 붙어있어야 함. k는 container를 들고있지 않

으며, c1은 p 상여 있고 p의 가장 위에 있어야 함. c2위에 c1이 있어야 함.

effect: and( (holding ?k ?c1) (top ?c2 ?p) (not (in ?c1 ?p)) (not (top ?c1 ?p))

(not (on ?c1 ?c2)) (not (empty ?k)) )

k는 c1를 들고있고, p 가장 위에는 c2가 있게 됨. c1은 p위에 있지 않으며, 가장

위에도 있지 않음. c2위에 c1이 있지 않으며, k는 아무것도 들고있지 않게 됨.

put ?k ?l ?c1 ?c2 ?p: location l의 crane k가 pile p위에 container c2위에 c1을 놓음.

precondition: and( (belong ?k ?l) (attached ?p ?l) (holding ?k ?c1) (top ?c2 ?p) )

k는 l에 속해있어야 하며, p는 l에 붙어있어야 함. k는 c1를 들고있으며,

c2은 p의 가장 위에 있어야 함.

effect: and( (in ?c1 ?p) (top ?c1 ?p) (on ?c1 ?c2) (not (top ?c2 ?p)) (not (holding ?k ?c1)) (empty ?k) )

p에 c1이 있으며 가장 위에 있음. c2위에 c1이 있음. p 가장 위에 c2가 있지 않

게 되며, k는 c1를 들고 있지 않으며, k는 아무것도 들고있지 않게 됨.

## 2. Problem definition (problem-02.pddl)

정의된 문제는 다음과 같다.

### 2.1) Objects

1개의 robot, 2개의 location, 2개의 crane, 4개의 pile, 6개의 container가 있다. 각각 r1, l1, l2, k1, k2, p1, q1, p2, q2, ca, cb, cc, cd, ce, cf이다.

### 2.2) Initialize

(adjacent l1 l2) (adjacent l2 l1)

(attached p1 l1) (attached q1 l1) (attached p2 l2) (attached q2 l2)

(belong k1 l1) (belong k2 l2)

(in ca p1) (in cb p1) (in cc p1) (in cd q1) (in ce q1) (in cf q1)

(on ca pallet) (on cb ca) (on cc cb)

(on cd pallet) (on ce cd) (on cf ce)

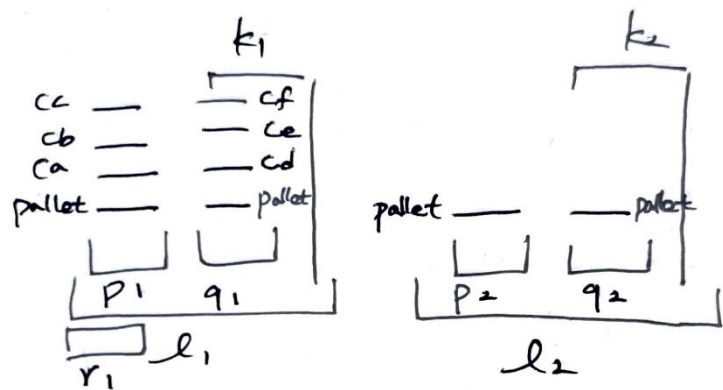
(top cc p1) (top cf q1)

(top pallet p2) (top pallet q2)

(at1 r1 l1) (unloaded r1) (free l2)

(empty k1) (empty k2))

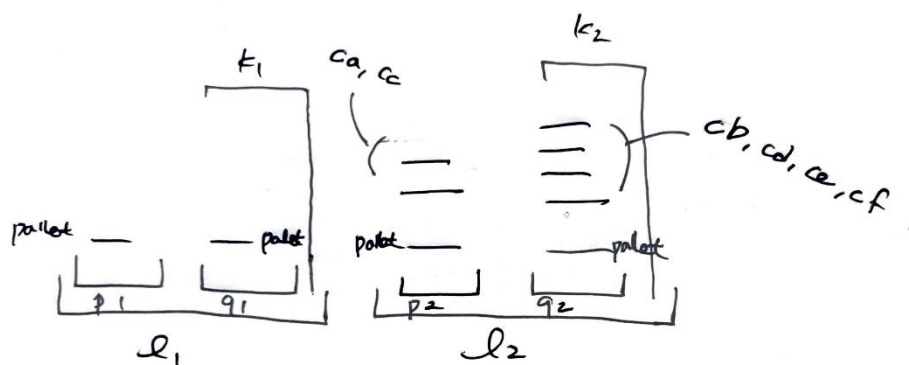
위 내용을 도식화하면 다음과 같다.



### 2.3) Goal

and( (in ca p2) (in cb q2) (in cc p2) (in cd q2) (in ce q2) (in cf q2) )

위 내용을 도식화하면 goal은 다음과 같다고 볼 수 있다.



### 3. Solve

Planner를 이용하 위 문제를 푼 결과 다음과 같다.

```

take k1 l1 cf ce q1
load k1 l1 cf r1
move r1 l1 l2
unload k2 l2 cf r1
put k2 l2 cf pallet q2
take k1 l1 cc cb p1
move r1 l2 l1
load k1 l1 cc r1
move r1 l1 l2
unload k2 l2 cc r1
put k2 l2 cc pallet p2
take k1 l1 ce cd q1
move r1 l2 l1
load k1 l1 ce r1
move r1 l1 l2
unload k2 l2 ce r1
put k2 l2 ce cf q2
take k1 l1 cd pallet q1
move r1 l2 l1
load k1 l1 cd r1
move r1 l1 l2
unload k2 l2 cd r1
put k2 l2 cd ce q2
take k1 l1 cb ca p1
move r1 l2 l1
load k1 l1 cb r1
move r1 l1 l2
unload k2 l2 cb r1
put k2 l2 cb cd q2
take k1 l1 ca pallet p1
move r1 l2 l1
load k1 l1 ca r1
move r1 l1 l2
unload k2 l2 ca r1
put k2 l2 ca cc p2
  
```

location	
l1	
l2	
object	
container	
robot	
location	
crane	
pile	
pile	
p1	
q1	
p2	
q2	
robot	
r1	
crane	
k1	
k2	
container	
pallet	
ca	
cb	
cc	
cd	
ce	
cf	

Solve 결과를 살펴보면 take move load move unload put를 반복으로 과정으로 이루어져 있으며, 6 번 반복하였다. 첫 반복에서는 이미 r1이 l1에 위치하여 있기 때문에 첫 move는 생략되었다.

각각의 반복은 cf, cc, ce, cd, cb, ca를 각각 q2, p2, q2, q2, q2, p2로 이동하는 과정으로 35 step만에 필요한 goal에 상태로 잘 이동된 것을 확인할 수 있었다.

#### 4. Additional Problem

만약 초기의 pile p1, p2의 쌓인 순서 그대로 q1, q2로 옮겨야 한다고 가정해보면 goal은 다음과 같이 정의할 수 있다.

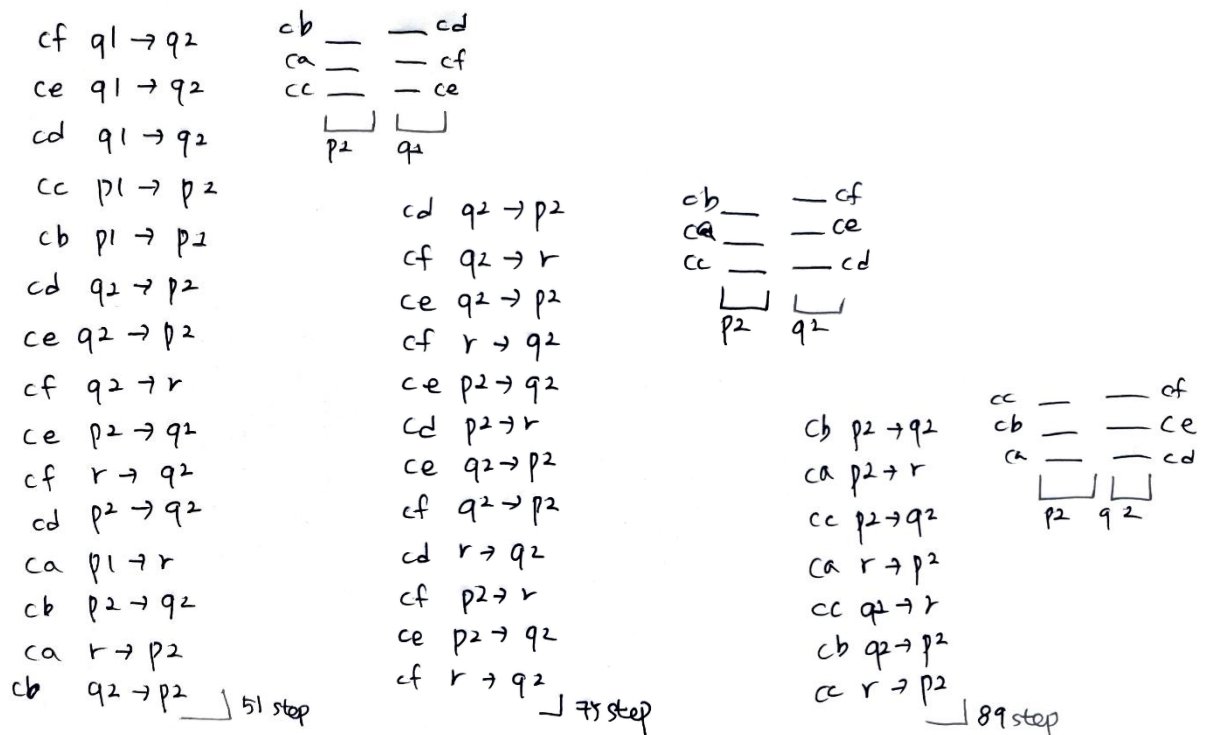
and( (in ca p2) (in cb p2) (in cc p2) (in cd q2) (in ce q2) (in cf q2)  
(on cb ca) (on cc cb) (on ce cd) (on cf ce) )

다음과 같이 goal state를 준 뒤 solve를 하니 89 step의 planning 결과가 나타났다. 결과를 분석해보면 크게 3개의 stage로 구성되어 있었다.

먼저 모든 container를 l2로 옮기는 과정이다. 총 51 step을 소모하였으며, 옮기는 도중에 순서를 정렬하기 편하게 robot에 임시 저장하여 crane을 통해 쌓인 순서를 바꾸는 과정도 포함되어 있었다.

모든 container를 p2, q2로 옮긴 이후에는 q2의 cd, ce, cf의 순서를 정렬하였다. 이는 75 step까지 진행되었으며, robot에 임시로 container 하나를 옮긴 뒤 crane을 이용하여 take put take put하는 과정으로 진행되었다.

마지막으로 p2의 container의 순서를 같은 방법으로 정렬하는 방법을 통해 89 step만에 p1과 q1에 있던 container를 순서 그대로 p2와 q2로 옮긴 것을 확인할 수 있었다.



```

| take k1 l1 cf ce q1
| load k1 l1 cf r1
| move r1 l1 l2
| unload k2 l2 cf r1
| put k2 l2 cf pallet q2
| move r1 l2 l1
| take k1 l1 ce cd q1
| load k1 l1 ce r1
| move r1 l1 l2
| unload k2 l2 ce r1
| put k2 l2 ce cf q2
| move r1 l2 l1
| take k1 l1 cd pallet q1
| load k1 l1 cd r1
| move r1 l1 l2
| unload k2 l2 cd r1
| put k2 l2 cd ce q2
| move r1 l2 l1
| take k1 l1 cc cb p1
| load k1 l1 cc r1
| move r1 l1 l2
| unload k2 l2 cc r1
| put k2 l2 cc pallet p2
| take k1 l1 cb ca p1
| move r1 l2 l1
| load k1 l1 cb r1
| move r1 l1 l2
| unload k2 l2 cb r1
| put k2 l2 cb cc p2
| take k2 l2 cd ce q2
| put k2 l2 cd cb p2
| take k2 l2 ce cf q2
| put k2 l2 ce cd p2
| take k2 l2 cf pallet q2
| load k2 l2 cf r1
| take k2 l2 ce cd p2
| put k2 l2 ce pallet q2
| unload k2 l2 cf r1
| put k2 l2 cf ce q2
| take k2 l2 cd cb p2
| put k2 l2 cd cf q2
| take k1 l1 ca pallet p1
| move r1 l2 l1
| load k1 l1 ca r1
| move r1 l1 l2

```

```

| take k2 l2 cb cc p2
| put k2 l2 cb cd q2
| unload k2 l2 ca r1
| put k2 l2 ca cc p2
| take k2 l2 cb cd q2
| put k2 l2 cb ca p2
| take k2 l2 cd cf q2
| put k2 l2 cd cb p2
| take k2 l2 cf ce q2
| load k2 l2 cf r1
| take k2 l2 ce pallet q2
| put k2 l2 ce cd p2
| unload k2 l2 cf r1
| put k2 l2 cf pallet q2
| take k2 l2 ce cd p2
| put k2 l2 ce cf q2
| take k2 l2 cd cb p2
| load k2 l2 cd r1
| take k2 l2 ce cf q2
| put k2 l2 ce cb p2
| take k2 l2 cf pallet q2
| put k2 l2 cf ce p2
| unload k2 l2 cd r1
| put k2 l2 cd pallet q2
| take k2 l2 cf ce p2
| load k2 l2 cf r1
| take k2 l2 ce cb p2
| put k2 l2 ce cd q2
| unload k2 l2 cf r1
| put k2 l2 cf ce q2
| take k2 l2 cb ca p2
| put k2 l2 cb cf q2
| take k2 l2 ca cc p2
| load k2 l2 ca r1
| take k2 l2 cc pallet p2
| put k2 l2 cc cb q2
| unload k2 l2 ca r1
| put k2 l2 ca pallet p2
| take k2 l2 cc cb q2
| load k2 l2 cc r1
| take k2 l2 cb cf q2
| put k2 l2 cb ca p2
| unload k2 l2 cc r1
| put k2 l2 cc cb p2

```

location	
l1	
l2	
object	
container	
robot	
location	
crane	
pile	
pile	
p1	
q1	
p2	
q2	
robot	
r1	
crane	
k1	
k2	
container	
pallet	
ca	
cb	
cc	
cd	
ce	
cf	