gripper-problem-2.pddl

```
(define (problem gripper-2)
(:domain gripper-domain)
(:objects rooma roomb roomc ball1 ball2 ball3 left right up down)
(:init
    (room rooma)
    (room roomb)
    (room roomc)
    (ball ball1)
    (ball ball2)
    (ball ball3)
    (gripper left)
    (gripper right)
    (gripper up)
    (gripper down)
    (at-robby rooma)
    (free left)
    (free right)
    (free up)
    (free down)
    (at ball1 rooma)
    (at ball2 roomb)
    (at ball3 roomc)
)
(:goal (and(at ball1 rooma) (at ball2 rooma) (at ball3 roomb) )))
```

1)

Found Plan (output)

(move rooma roomb)
(pick ball2 roomb left)
(move roomb rooma)
(drop ball2 rooma left)
(move rooma roomc)
(pick ball3 roomc left)
(move roomc roomb)
(drop ball3 roomb left)

```
(:action move
  :parameters (rooma roomb)
  :precondition
    (and
      (room rooma)
      (room roomb)
      (at-robby rooma)
    )
  :effect
    (and
      (at-robby roomb)
      (not
        (at-robby rooma)
      )
    )
)
```

2)

(move rooma roomb)

(pick ball2 roomb left)

(move roomb rooma)

(drop ball2 rooma left)

(move rooma roomc)

(pick ball3 roomc left)

(move roomc roomb)

(drop ball3 roomb left)

```
(:action pick
  :parameters (ball2 roomb left)
  :precondition
    (and
      (ball ball2)
      (room roomb)
      (gripper left)
      (at ball2 roomb)
      (at-robby roomb)
      (free left)
    )
  :effect
    (and
      (carry ball2 left)
      (not
        (at ball2 roomb)
      )
      (not
        (free left)
      )
    )
)
```

3)

(move rooma roomb)

(pick ball2 roomb left)

(move roomb rooma)

(drop ball2 rooma left)

(move rooma roomc)

(pick ball3 roomc left)

(move roomc roomb)

(drop ball3 roomb left)

```
(:action move
  :parameters (roomb rooma)
  :precondition
    (and
      (room roomb)
      (room rooma)
      (at-robby roomb)
    )
  :effect
    (and
      (at-robby rooma)
      (not
        (at-robby roomb)
      )
    )
)
```

4)

(move rooma roomb)
(pick ball2 roomb left)
(move roomb rooma)
(drop ball2 rooma left)
(move rooma roomc)
(pick ball3 roomc left)
(move roomc roomb)
(drop ball3 roomb left)

```
(:action drop
  :parameters (ball2 rooma left)
  :precondition
    (and
      (ball ball2)
      (room rooma)
      (gripper left)
      (carry ball2 left)
      (at-robby rooma)
    )
  :effect
    (and
      (at ball2 rooma)
      (free left)
      (not
        (carry ball2 left)
      )
    )
)
```

5)

(move rooma roomb)
(pick ball2 roomb left)
(move roomb rooma)
(drop ball2 rooma left)
(move rooma roomc)
(pick ball3 roomc left)
(move roomc roomb)
(drop ball3 roomb left)

```
(:action move
  :parameters (rooma roomc)
  :precondition
    (and
      (room rooma)
      (room roomc)
      (at-robby rooma)
    )
  :effect
    (and
      (at-robby roomc)
      (not
        (at-robby rooma)
      )
    )
)
```

6)

Steps
(move rooma roomb)
(pick ball2 roomb left)
(move roomb rooma)
(drop ball2 rooma left)
(move rooma roomc)
(pick ball3 roomc left)
(move roomc roomb)
(drop ball3 roomb left)

```
(:action pick
  :parameters (ball3 roomc left)
  :precondition
    (and
      (ball ball3)
      (room roomc)
      (gripper left)
      (at ball3 roomc)
      (at-robby roomc)
      (free left)
    )
  :effect
    (and
      (carry ball3 left)
      (not
        (at ball3 roomc)
      )
      (not
        (free left)
      )
    )
)
```

7)

Steps
(move rooma roomb)
(pick ball2 roomb left)
(move roomb rooma)
(drop ball2 rooma left)
(move rooma roomc)
(pick ball3 roomc left)
(move roomc roomb)
(drop ball3 roomb left)

```
(:action move
  :parameters (roomc roomb)
  :precondition
    (and
      (room roomc)
      (room roomb)
      (at-robby roomc)
    )
  :effect
    (and
      (at-robby roomb)
      (not
        (at-robby roomc)
      )
    )
)
```

8)

(move rooma roomb)

(pick ball2 roomb left)

(move roomb rooma)

(drop ball2 rooma left)

(move rooma roomc)

(pick ball3 roomc left)

(move roomc roomb)

(drop ball3 roomb left)

```
(:action drop
  :parameters (ball3 roomb left)
  :precondition
    (and
      (ball ball3)
      (room roomb)
      (gripper left)
      (carry ball3 left)
      (at-robby roomb)
    )
  :effect
    (and
      (at ball3 roomb)
      (free left)
      (not
        (carry ball3 left)
      )
    )
)
```