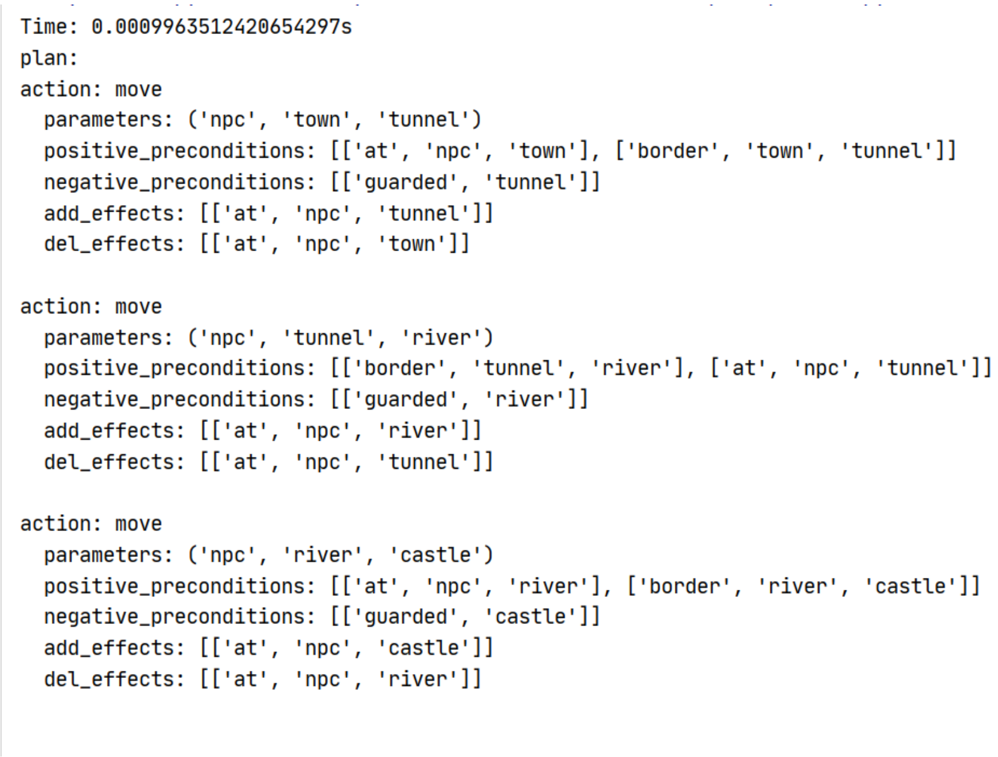
The result:



```
Time: 0.0009963512420654297s
plan:
action: move
  parameters: ('npc', 'town', 'tunnel')
  positive_preconditions: [['at', 'npc', 'town'], ['border', 'town', 'tunnel']]
  negative_preconditions: [['guarded', 'tunnel']]
  add_effects: [['at', 'npc', 'tunnel']]
  del_effects: [['at', 'npc', 'town']]

action: move
  parameters: ('npc', 'tunnel', 'river')
  positive_preconditions: [['border', 'tunnel', 'river'], ['at', 'npc', 'tunnel']]
  negative_preconditions: [['guarded', 'river']]
  add_effects: [['at', 'npc', 'river']]
  del_effects: [['at', 'npc', 'tunnel']]

action: move
  parameters: ('npc', 'river', 'castle')
  positive_preconditions: [['at', 'npc', 'river'], ['border', 'river', 'castle']]
  negative_preconditions: [['guarded', 'castle']]
  add_effects: [['at', 'npc', 'castle']]
  del_effects: [['at', 'npc', 'river']]
```

The result is mean that 2 steps needed to get the goal state. First move npc from town to tunnel, second move npc from tunnel to river, finally more npc from river to castle.

To solve the Wumpus World problem:

First, define the Wumpus world.

```
(define (domain wumpus-world)
   (:requirements :strips :typing)
   (:types player location monster element chest)
(:predicates
       (adj ?square-1 ?square-2)
       (pit ?square)
       (at ?what ?square)
       (have ?who ?what)
       (dead ?who))
   (:action move
     :parameters (?p - player ?l1 - location ?l2 - location)
     :precondition (and (at ?p ?l1)
                           (border ?l1 ?l2)
                           (not (guarded ?l2)))
     :effect (and (at ?p ?l2) (not (at ?p ?l1)))
   )
  (:action move
       :parameters (?who ?from ?to)
       :precondition (and (adj ?from ?to)
                           (not (pit ?to))
                           (at ?who ?from))
       :effect (and (not (at ?who ?from))
                           (at ?who ?to))
  )
  (:action take
       :parameters (?who ?what ?where)
       :precondition (and (at ?who ?where)
                           (at ?what ?where))
       :effect (and (have ?who ?what)
                    (not (at ?what ?where)))
  )
  (:action shoot
       :parameters (?who ?where ?arrow ?victim ?where-
  victim)
       :precondition (and (have ?who ?arrow)
                           (at ?who ?where)
                           (at ?victim ?where-victim)
                           (adj ?where ?where-victim))
       :effect (and (dead ?victim) (not (at ?victim ?where-
  victim)) (not (have ?who ?arrow)))
  )
)
```

Second, define the problem:

```
  (define (problem Wumpus-move)
       (:domain wumpus-world)
       (:objects
             sq-1-1 sq-1-2 sq-1-3
             sq-2-1 sq-2-2 sq-2-3
             the-gold
             the-arrow
             agent
             wumpus)
       (:init
              (adj sq-1-1 sq-1-2) (adj sq-1-2 sq-1-1)
              (adj sq-1-2 sq-1-3) (adj sq-1-3 sq-1-2)
              (adj sq-2-1 sq-2-2) (adj sq-2-2 sq-2-1)
              (adj sq-2-2 sq-2-3) (adj sq-2-3 sq-2-2)
              (adj sq-1-1 sq-2-1) (adj sq-2-1 sq-1-1)
              (adj sq-1-2 sq-2-2) (adj sq-2-2 sq-1-2)
              (adj sq-1-3 sq-2-3) (adj sq-2-3 sq-1-3)
              (pit sq-1-2)
              (at the-gold sq-1-3) (at agent sq-1-1)
              (have agent the-arrow) (at wumpus sq-2-3))
       (:goal (and (have agent the-gold)(at agent sq-1-1)))
  )
```

Then solve the PDDL.