

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action moveto
:parameters (red home woods)
:precondition
  (and
    (forward red woods)
    (at red home)
    (not
      (stop red)
    )
  )
:effect
  (and
    (at red woods)
    (not
      (at red home)
    )
  )
)
```

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action tell-path
:parameters (red wolf woods)
:precondition
  (and
    (at red woods)
    (at wolf woods)
    (not
      (= red wolf)
    )
    (not
      (path_know red)
    )
    (path_know wolf)
  )
:effect
  (and
    (path_know red)
    (stop red)
    (not
      (stop wolf)
    )
    (hungry wolf)
  )
)
```

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action moveto
:parameters (wolf woods grannyhouse)
:precondition
  (and
    (forward wolf grannyhouse)
    (at wolf woods)
    (not
      (stop wolf)
    )
  )
:effect
  (and
    (at wolf grannyhouse)
    (not
      (at wolf woods)
    )
  )
)
```

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action eat-alive
:parameters (wolf granny grannyhouse)
:precondition
  (and
    (at wolf grannyhouse)
    (at granny grannyhouse)
    (hungry wolf)
    (not
      (= wolf granny)
    )
    (not
      (stop granny)
    )
    (alive wolf)
    (alive granny)
  )
:effect
  (and
    (eaten granny)
  )
)
```

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action pick-up
:parameters (red flower woods)
:precondition
  (and
    (at red woods)
    (stop red)
    (like red flower)
  )
:effect
  (and
    (have red flower)
    (not
      (stop red)
    )
  )
)
```

Found Plan (output)

(moveto red home woods)

(tell-path red wolf woods)

(moveto wolf woods grannyhouse)

(eat-alive wolf granny grannyhouse)

(pick-up red flower woods)

(moveto red woods grannyhouse)

(eat-alive wolf red grannyhouse)

(sleep-snoreloud wolf granny red grannyhouse)

```
(:action moveto
:parameters (red woods grannyhouse)
:precondition
  (and
    (forward red grannyhouse)
    (at red woods)
    (not
      (stop red)
    )
  )
:effect
  (and
    (at red grannyhouse)
    (not
      (at red woods)
    )
  )
)
```

Found Plan (output)

(moveto red home woods)
(tell-path red wolf woods)
(moveto wolf woods grannyhouse)
(eat-alive wolf granny grannyhouse)
(pick-up red flower woods)
(moveto red woods grannyhouse)
(eat-alive wolf red grannyhouse)
(sleep-snoreloud wolf granny red grannyhouse)

```
(:action eat-alive
:parameters (wolf red grannyhouse)
:precondition
  (and
    (at wolf grannyhouse)
    (at red grannyhouse)
    (hungry wolf)
    (not
      (= wolf red)
    )
    (not
      (stop red)
    )
    (alive wolf)
    (alive red)
  )
:effect
  (and
    (eaten red)
  )
)
```

Found Plan (output)

(moveto red home woods)
(tell-path red wolf woods)
(moveto wolf woods grannyhouse)
(eat-alive wolf granny grannyhouse)
(pick-up red flower woods)
(moveto red woods grannyhouse)
(eat-alive wolf red grannyhouse)
(sleep-snoreloud wolf granny red grannyhouse)

```
(:action sleep-snoreloud
:parameters (wolf granny red grannyhouse)
:precondition
  (and
    (at wolf grannyhouse)
    (at granny grannyhouse)
    (at red grannyhouse)
    (hungry wolf)
    (eaten granny)
    (eaten red)
  )
:effect
  (and
    (snore wolf)
    (not
      (hungry wolf)
    )
  )
)
```