

The diagram illustrates a state transition system for a robot's behavior based on three variables: hunger, thirst, and fatigue (tired). The states are represented by ovals: **Resting**, **drinking**, **Sleeping**, and **Nothing/other**.

Transitions and Conditions:

- Resting to drinking:** $thirst > n$ and $hunger < x$
- drinking to Resting:** $thirst \leq x$ and $hunger > n$
- Resting to Sleeping:** $hunger > x$ and $thirst \leq x$
- Sleeping to Resting:** $tired < x$
- Sleeping to Nothing/other:** $tired > n$
- Nothing/other to Resting:** $thirst < n$ and $hunger < x$
- Nothing/other to drinking:** $thirst > x$

Variables: hunger, thirst, Fatigue (i.e. tired)

```
class SimpleFSM:
    def Thinking(self):
        #local variables
        fatigue = 0
        hunger = 0
        thirst = 0

        states = ["eating", "drinking", "sleeping", "awake"]
        currentState = "awake"

        alive = True
        running = True
        maxLimit = 100
        gameTime = 0

        while running and alive:
            gameTime += 1

            #sleeping: reduce fatigue, everything else increases
            if currentState is "sleeping":
                #sleep
                print("Zzzzzzzzzzz")
                fatigue -= 1
                hunger += 0.5
```

```

thirst += 0.5

#check if not tired
if fatigue < 5:
    #check for other states
    if thirst > 7:
        currentState = "drinking"
    elif hunger > 7:
        currentState = "eating"
    else:
        currentState = "awake"

#awake: doing nothing, all variables increase
elif currentState is "awake":
    #do nothing
    print("Bored . . .")
    hunger += 1
    thirst += 1
    fatigue += 1

    #check for other states
    if fatigue > 15:
        currentState = "sleeping"
    elif thirst > 7:
        currentState = "drinking"
    elif hunger > 7:
        currentState = "eating"

#eating: hunger reduces, fatigue and thirst increase
elif currentState is "eating":
    #eat
    print("Om nom nom nom")
    hunger -= 2
    thirst += 1
    fatigue += 0.5

    #check for other states
    if fatigue > 15:
        currentState = "sleeping"
    elif thirst > 7:
        currentState = "drinking"
    elif hunger < 2:
        currentState = "awake"

#drinking: thirst reduces, all other states increase
elif currentState is "drinking":
    #drink
    print("Gulp gulp gulp")
    thirst -= 2
    hunger += 1
    fatigue += 0.5

    #check for other states
    if fatigue > 15:
        currentState = "sleeping"
    elif thirst < 2:
        if hunger > 7:
            currentState = "eating"
        else:
            currentState = "awake"
    else:

```

```

        #broken
        print("Why are you here? The code must have something wrong with it .
. .")
        die()

#checking if starved to death
if hunger > 20:
    alive = False
    print("Gurgle . . . Gurgle . . . *death rattle*")

elif thirst > 20:
    alive = False
    print("*gasp* . . . *gasp* . . . *death rattle*")

elif fatigue > 20:
    alive = False
    print("*thud* . . . *death rattle*")

#checking for end of game time
if gameTime > maxLimit:
    running = False
    print("*siren goes off*")

print("Hello world!")

ai = SimpleFSM()

ai.Thinking()

print("--- The End ---")

```