```
monitor Barber_Shop {
  int barber = 0, chair = 0, open = 0;
  cond barber_available;    # signaled when barber > 0
  cond chair_occupied;      # signaled when chair > 0
  cond door_open;           # signaled when open > 0
  cond customer_left;       # signaled when open == 0
  procedure get_haircut() {
    while (barber == 0) wait(barber_available);
    barber = barber - 1;
    chair = chair + 1; signal(chair_occupied);
    while (open == 0) wait(door_open);
    open = open - 1; signal(customer_left);
  }
  procedure get_next_customer() {
    barber = barber + 1; signal(barber_available);
    while (chair == 0) wait(chair_occupied);
    chair = chair - 1;
  }
  procedure finished_cut() {
    open = open + 1; signal(door_open);
    while (open > 0) wait(customer_left);
  }
}
```

Figure 5.10 Sleeping barber monitor.