```python
# Ground-Truth Preconditions

def goto(self, recep):
  assert self.is_visible(recep)

def open(self, recep):
  assert self.is_visible(recep)
  assert self.env.get_property(recep, 'open') == False

def close(self, recep):
  assert self.is_visible(recep)
  assert self.env.get_property(recep, 'open') == True

def take(self, obj, recep):
  assert self.is_visible(obj)
  assert self.is_visible(recep)
  assert self.inventory == None

def put(self, obj, recep):
  assert self.inventory == obj
  assert self.is_visible(recep)

def clean(self, obj, recep):
  assert self.is_visible(recep)
  assert self.inventory == obj
  assert 'sink' in recep

def heat(self, obj, recep):
  assert self.is_visible(recep)
  assert self.inventory == obj
  assert 'microwave' in recep

def cool(self, obj, recep):
  assert self.is_visible(recep)
  assert self.inventory == obj
  assert 'fridge' in recep

def toggle(self, obj):
  assert self.is_visible(obj)
```

```python
# Predicted Preconditions

 def goto(self, recep):
  assert self.is_visible(recep)

def open(self, recep):
  assert recep in self.visible_objects
  assert self.env.get_property(recep, 'open') is False

def close(self, recep):
  assert self.env.get_property(recep, 'open')

def take(self, obj, recep):
  assert self.is_visible(obj)
  assert self.inventory is None

def put(self, obj, recep):
  assert obj in self.inventory

def clean(self, obj, recep):
  assert obj in self.inventory

def heat(self, obj, recep):
  assert self.inventory != None
  assert self.env.get_property(obj, 'heat') == False

def cool(self, obj, recep):
  assert obj in self.inventory
  assert self.env.get_property(obj, 'cool') == False

def toggle(self, obj):
  assert self.is_visible(obj)
  assert self.inventory is not None
```