

# 6章ステートマシン図

オブジェクトには状態という概念がある。オブジェクトの状態は、あるきっかけ(イベント)が起こった際に、別の状態になる。この状態の移り変わりを遷移と呼ぶ。

ステートマシン図は、オブジェクトが存在している期間の状態(ライフサイクル、生存期間)とその移り変わり(状態遷移)を表す図である。

UMLでは、状態の変化を把握する必要のあるオブジェクトに限定して分析を行う。

オブジェクトは、前の状態が終了した場合やイベントを受け取った場合に状態遷移します。

状態遷移は、ガードやアクションを伴うことがあり、ガードは状態遷移の条件となる。アクションは、状態遷移の前に実行する短い処理である。

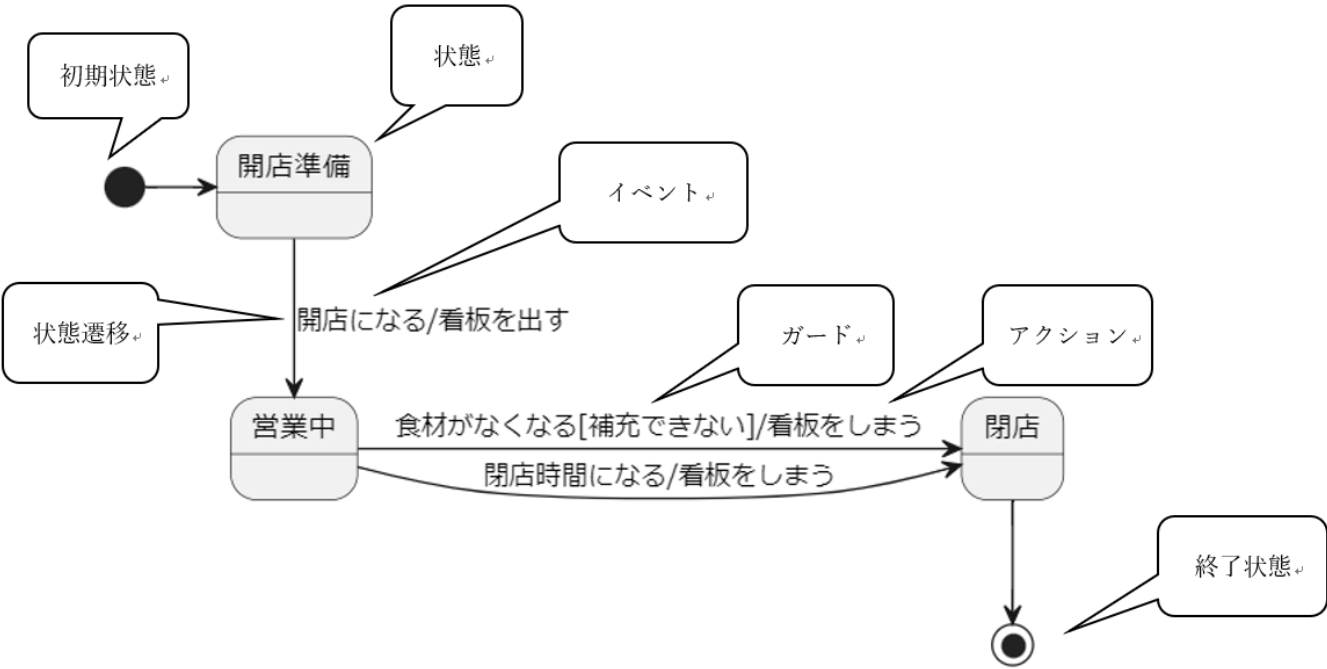
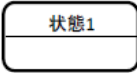
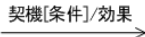


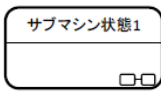







図5-2 ステートマシン図

▼ステートマシンの要素

要素	表示形式	意味
開始状態		状態遷移の開始
終了状態		状態遷移の終了

状態		実態の状態 (機能は省略可能)
遷移		状態の遷移を示す矢印。契機、条件、効果はそれぞれ省略可
選択点		前の状態が、遷移条件を示すガードに合致する方へ状態遷移する。 複数の条件に合致した場合は、その遷移先は任意となり、合致するものが無い場合は記述の誤りとみなす。
連結点		二つ以上の遷移が集約された状態、または、二つ以上の遷移が離散する状態、もしくはその両方が起こる状態
コンポジット状態		合成状態の略式表記
入場点		合成状態の入り口
退場点		合成状態の出口
停止状態		停止状態は、状態マシン図で定義されている事象そのものが完了することを表す
浅い履歴状態		以前状態に戻ることを表す。浅い履歴状態では、形式的にもどることを意味しており、実際に遷移前と遷移後で以前の状態に変化あっても考慮しない
深い履歴状態		以前状態にもどることを表す。深い履歴状態では、実際に遷移前と同じ状態に戻る

## ▼ステートマシン図の例

### ・基本構成

希望の時間にご飯を炊く、炊飯器。

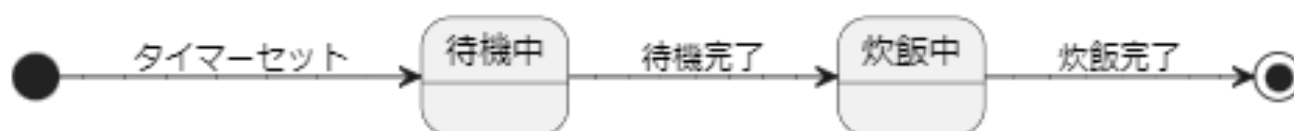


図5-3 基本構成

@startuml 基本構成  
left to right direction

```

[*] --> 待機中 : タイマーセット
待機中 --> 炊飯中 : 待機完了
炊飯中 --> [*] : 炊飯完了

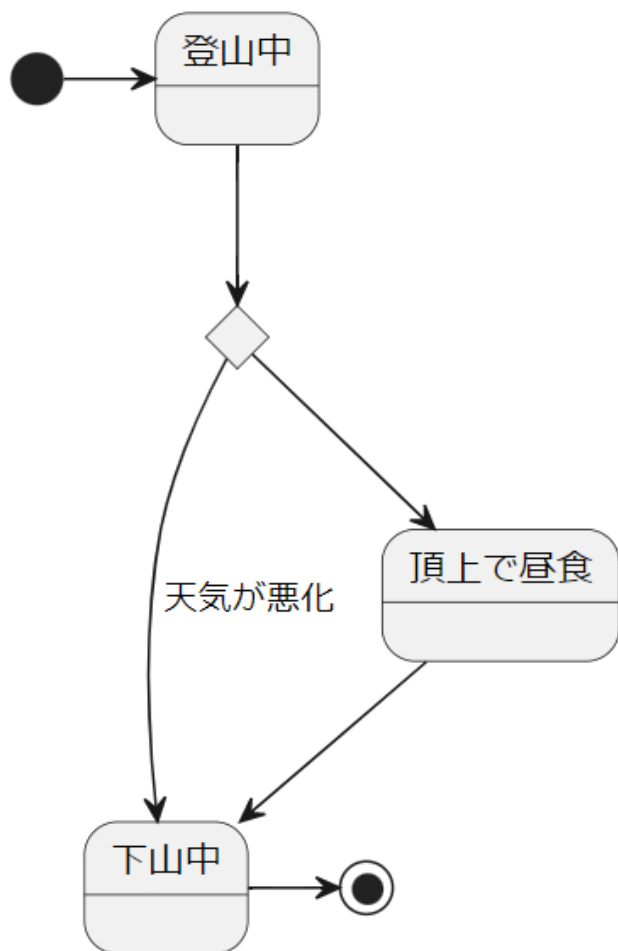
```

```
@endum1
```

図 5-3 基本構成

#### ・選択点と結合点

登山中に天候が悪化したら、登頂をあきらめて下山する



PlantUMLでの表記

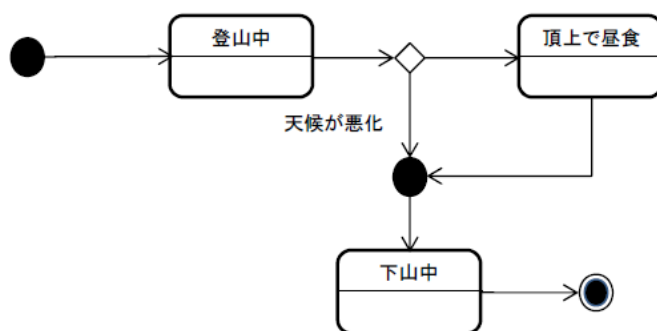


図5-4 選択点と結合点

```

@startuml
' left to right direction
[*] -right-> 登山中
state c <<choice>>
state c <<choice>>
登山中 -down-> c
c --> 頂上で昼食
c -right-> 下山中 : 天気が悪化
' PlantUMLに結合点の表示形式はない

```

頂上で昼食 --> 下山中

下山中 -right-> [\*]

@endum1

#### ・入場点と退場点

自動車で通勤中の状態を入場点と退場点で表す。この自動車で通勤中の状態の中には、停止中と運転中の状態がある。なお、自動車に乗るときと降りるときは停止中である。

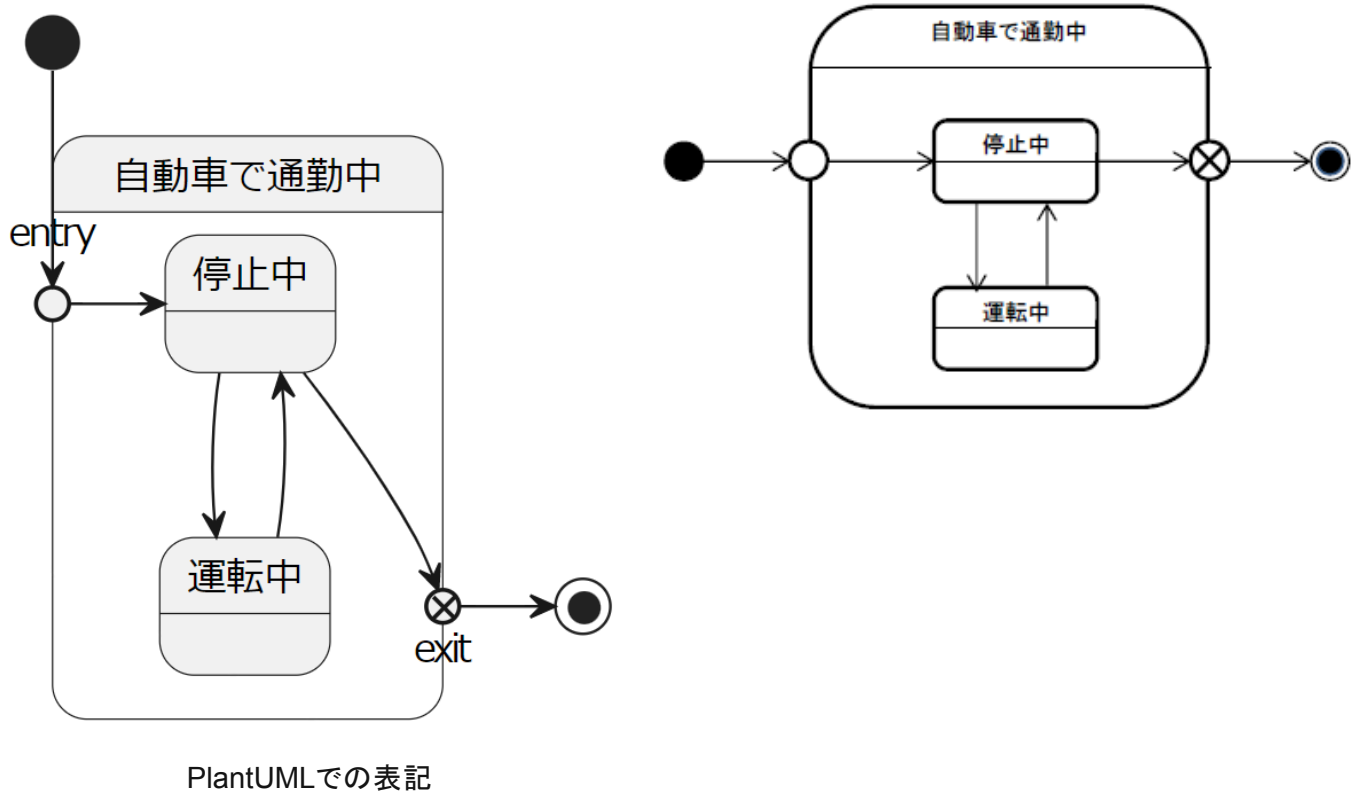


図 5-5 入場点と退場点

@startuml

```
state 自動車通勤中 {  
    state entry <<entryPoint>>  
    state exit <<exitPoint>>
```

```
entry -> 停止中
```

```
停止中 -down-> 運転中
```

```
運転中 -up-> 停止中
```

```

    停止中 -right-> exit
}
[*] --> entry
exit -> [*]

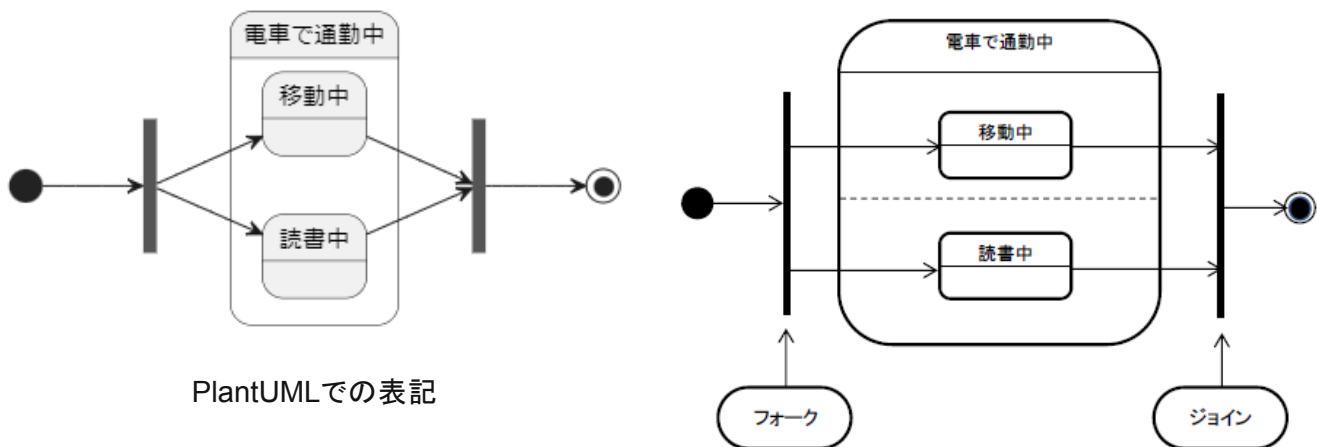
@enduml

```

## ・直交状態とフォーク・ジョイン

電車で通勤中は、移動する状態と読書する状態が互いに影響を及ぼさないので、直交状態という。

二つの状態は破線で区切って表示する。また、この直交状態の始まりと終わりはフォークとジョインで表すことができる。



今回の例では同時状態 (-----) の表示ができなかったが、記号 `--` または `||` で分離することで、合成状態の中に同時状態を定義することができる。

## 演習6—1

(1) あるプリンターには「待機中」「印刷中」「用紙無し」の各状態があります。これらの状態の遷移をステートマシン図で描いてください。

【提出ファイル】演習6-1\_SD\_プリンタ.pu

# 演習6ー2

部屋オブジェクトには、空室状態と、予約済状態、使用状態がある。各状態のイベントや状態の入れ子について考察し、部屋オブジェクトのステートマシン図を作成しなさい。

【提出ファイル】演習6-2\_SD\_部屋オブジェクト.pu

# 演習6ー3

次の病院の診察予約システムについて、ユースケース図とステートマシン図を作成しなさい。

○ユースケース図・・・アクターは2人です。

【提出ファイル】演習6-3\_ユースケース図\_診察予約システム.pu

○ステートマシン図・・・スマートフォンの4つの画面をオブジェクトとします。

【提出ファイル】演習6-3\_ステートマシン図\_診察予約システム.pu

【シナリオ】

スマートフォンの画面は次の4画面です。



〇〇医院診察予約

あなたの順番

【 9 】

メニューへ

〇〇医院診察予約

今の順番

【 5 】

メニューへ

- ① 予約システムに接続すると、「メニュー」画面が表示され、各機能へリンクする。
- ② 予約をする場合は「予約と確認」画面では、名前と電話番号を入力して、登録ボタンを押す。
- ③ 登録ボタンを押すと、システムには、現在の登録者番号の次に、名前、電話番号、が記録される。そして、「あなたの順番」画面へリンクする。
- ④ 登録した自分の順番を確認する場合は、「予約と確認」画面で、名前か電話番号を入力してから確認ボタンを押すと、「あなたの順番」画面へリンクする。
- ⑤ 登録した自分の順番を取り消す場合は、「予約と確認」画面で、名前か電話番号を入力してから取消ボタンを押すと、「メニュー」画面へリンクする。
- ⑥ 「あなたの順番」画面ではあなたの登録者番号を表示する。
- ⑦ 「今の順番」画面では、現在診察中の患者の順番を表示する。診察が終わると看護師が現在の患者番号を一つ増やす。

## 演習6－4

---

次のソフトドリンクの自動販売機について、各図を作成してください。

### 問題1

「自動販売機」「ボタン」「料金装置」「ベンダー装置」をクラスとして、クラス図を作成してください。

【提出ファイル】演習**6-4-1\_クラス図\_自動販売機.pu**

## 問題2

客とベンダーから見た自動販売機の機能を、ユースケース図で作成してください。

【提出ファイル】演習**6-4-2\_ユースケース図\_自動販売機.pu**

## 問題3

客がドリンクを手にするまでの手順をシーケンス図で作成してください。

【提出ファイル】演習**6-4-3\_シーケンス図\_自動販売機.pu**

## 問題4

客がドリンクを手にするまでの手順をコミュニケーション図で作成してください。

【提出ファイル】演習**6-4-4\_コミュニケーション図\_自動販売機.pu**

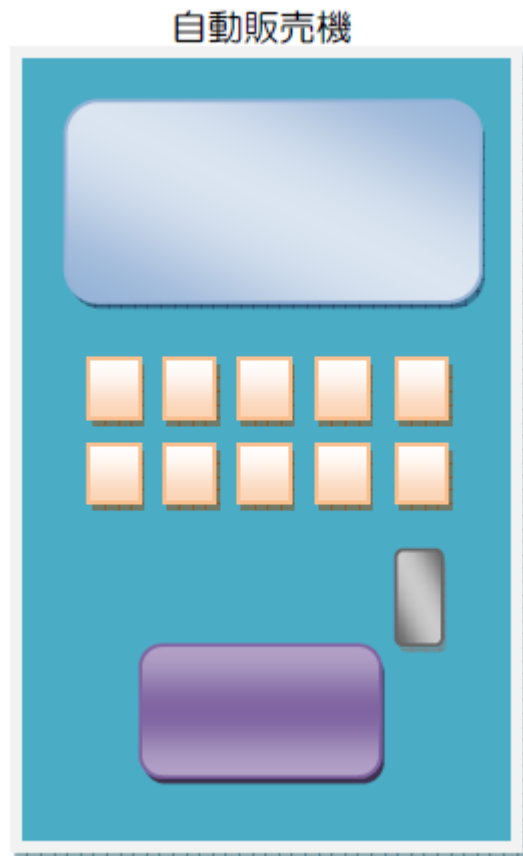
## 問題5

客がドリンクを手にするまでの手順をステートマシン図で作成してください。

【提出ファイル】演習**6-4-5\_ステートマシン図\_自動販売機.pu**

【シナリオ】





ドリンク 10 種類



ボタン 10 種類

料金装置

ベンダー装置



客



ベンダー

- ① 自動販売機は、10 個のボタン、料金装置、ベンダー装置からできている。
- ② 10 種類のドリンクは、残りが半分以下になるとベンダーが運んでくる。
- ③ ②の時、ベンダーは集まった料金を回収する。
- ④ 客はお金を料金装置に入れる。(おつりは考えなくて良い)
- ⑤ ボタンを押す。
- ⑥ ベンダー装置から、ドリンクが出てくる。

## 演習6－5

次のライトレーサーについて、次の各図を作成してください。

### 問題1

「ライトレーサー」「光センサー」「右モータ」「左モータ」「CPU」「電源SW」についてクラス図を作成してください。

【提出ファイル】演習6-5-1\_クラス図\_ライトレーサー.pu

## 問題2

ユースケース図を作成してください。

ヒント ユーザーが求める機能は1つです。また、ユーザーの他にもアクターになる部品があります。

【提出ファイル】演習6-5-2\_ユースケース図\_ライトレーサー.pu

## 問題3

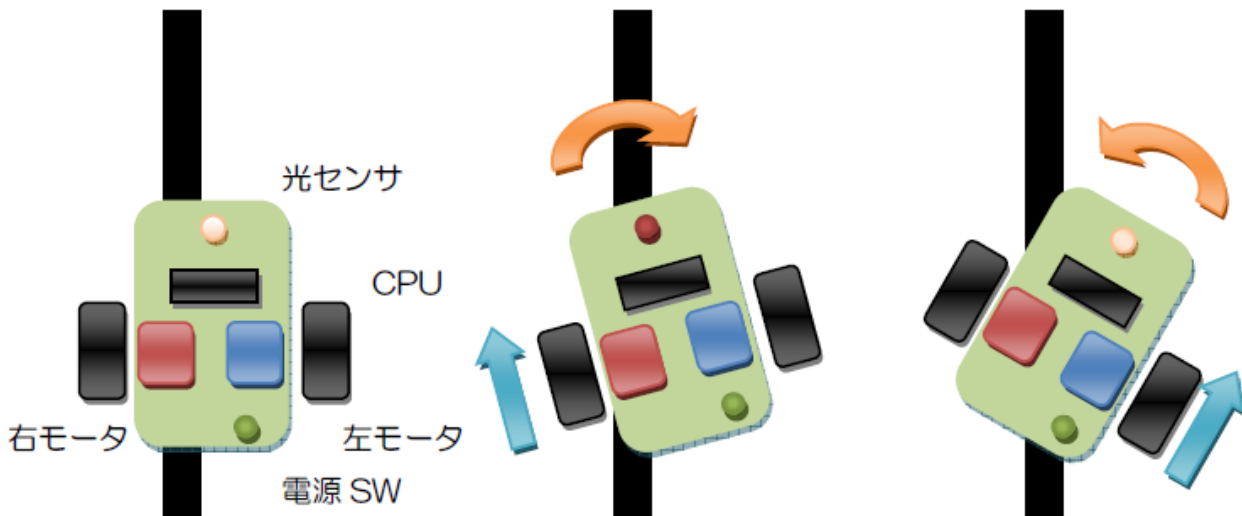
ステートマシン図を作成してください。

ヒント ① ライトレーサーが動作しているときの状態図を描きます。(状態は2つです)

② 電源SW がON・OFF の状態図を描きますが、ON の状態は①の図が入れ子になります。

【提出ファイル】演習6-5-3\_ステートマシン図\_ライトレーサー.pu

## 【シナリオ】



① このライトレーサーは、光センサー1個、左右モータ、CPU、電源SW(スイッチ)から出来ている。

② 電源SW を入れるとシステムが起動する。

③ 光センサーが黒いライン上にあると、「暗い」と判断し、右モータOFF、左モータON にする。したがって、ライトレーサーは右旋回を始める。

④ 光センサーがラインを外れると、「明るい」と判断し、右モータON、左モータOFF にする。

したがって、ライトレーサーは左旋回を始める。

⑤ ライトレーサーはラインの右境界線上を進む。

⑥ 電源SW を切るとシステムが停止する。