

# プログラミング応用

<http://bit.ly/ouyou3d>

## 計算モデル (2)

後期 第6週

2018/11/1

本日は・・・

オートマトンで**複数の終了状態**を  
表せるように実装を改良します

# 「①②の処理」の改良

```
/* (1) 変数を宣言 */
```

```
char input[100];          //入力を格納する配列
```

```
int i=0;                  //入力数をカウントする
```

```
//状態は、0が奇数、1が偶数の状態を表す
```

```
int current_state=0; //現在の状態（初期状態）
```

```
int fin_state=1;        //終了状態
```

配列 **fin\_states** に変更する  
(終了状態が要素となった配列)

```
/* (2) 文字列の入力 */
```

```
printf("数字を入力してください。 \n");
```

```
scanf("%s", input);
```

# 「④の処理」の改良

```
/* (4) 終了状態にいるか判定 */  
if(current_state == fin_state) {  
    printf("受理する。 \n");  
} else {  
    printf("受理しない。 \n");  
}
```

「current\_stateが、配列fin\_states  
に含まれているかを判別する」関数  
isaccept()を作り、ここで呼び出す

まずは、先週の2\_05\_even.cにこれらの改良をしてみます

# 【課題6-1】

「2\_05\_even.c」の終了状態を表す変数fin\_stateを、配列fin\_statesに変更して下さい。ただし、fin\_statesの最後の要素数は必ず -1 とします。

```
//終了状態が格納された配列
```

```
int fin_states[2] = {1, -1};
```

# 【課題6-2】

「2\_05\_even.c」に、「current\_stateが、配列 fin\_statesに含まれているかを判別する」関数 isaccept()を作って、④の部分を改良して下さい。

[この関数のプロトタイプ宣言]

```
int isaccept(int c, int fin_states[]);
```

// [引数] c: 現在の状態    fin\_states: 終了状態の配列

// [戻り値] 0: 受理不可    1: 受理可能

- ▶ for文を使って、cがfin\_statesのi番目と等しいかどうかを調べ、等しかった場合、「受理可能」なので1を返す
- ▶ for文は「fin\_statesのi番目が -1 になるまで」繰り返す  
(fin\_statesは必ず最後の要素が -1 になるように宣言しておく)
- ▶ for文が終了した場合は、「終了状態に含まれていない、つまり受理できない」ため、0を返す

# 【課題6-2】

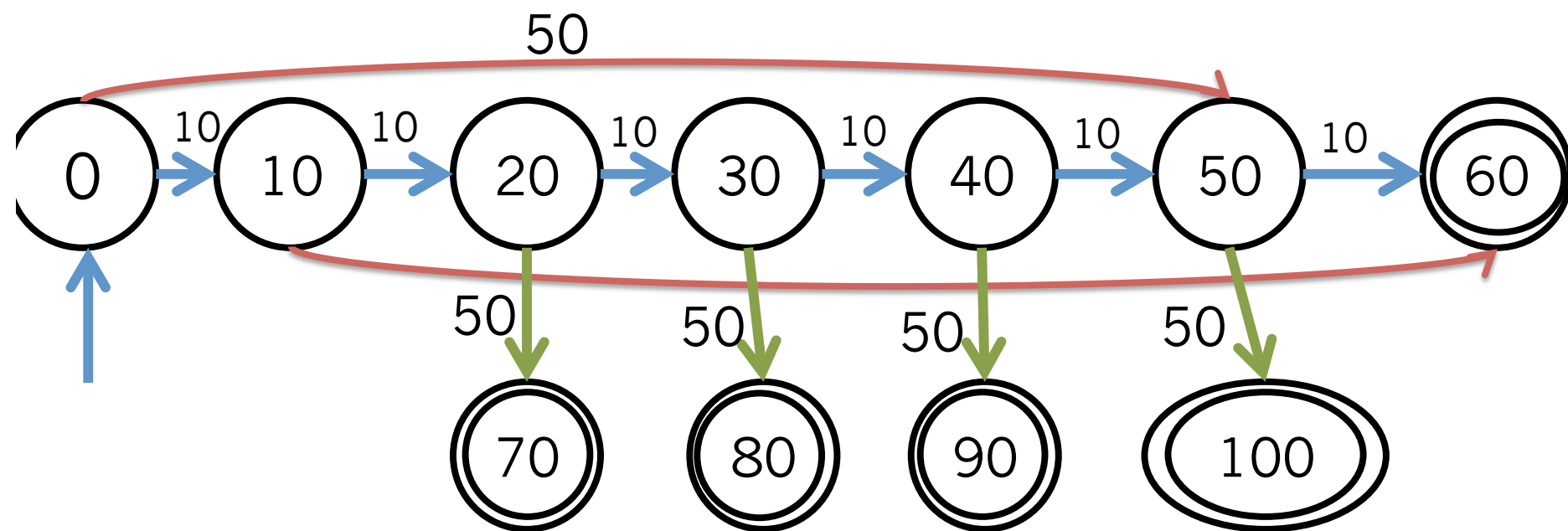
関数isaccept()を呼び出す箇所は以下のように変更します。

```
/* (4) 終了状態にいるか判定 */  
if(isaccept(current_state, fin_states)) {  
    printf("受理する。 \n");  
} else {  
    printf("受理しない。 \n");  
}
```

実行結果は、練習5-1と同じく偶数の入力は受理されます

# 【課題6-3】

前回の例で示した、自動販売機のオートマンを実装して下さい。



- ▶ 状態は、0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100とします
- ▶ 終了状態は、60, 70, 90, 100とします（最後の要素は-1にする）
- ▶ 入力は、10円はt、50円はf、または1、5など一文字とします



# 前期中間試験について

以下に関わるプログラミングの問題を出題します

- ▶ バブルソート、クイックソート
- ▶ ニュートン法
- ▶ オートマトン

# 小テストについて

## 小テストの注意点

- 他人の力は借りずに、自分だけでプログラムを作成する。（つまり定期試験と同様）
- 小テスト中は、演習室外へのネットワークアクセスは遮断される。

# 小テストについて

## 小テスト中に参照できるもの

- 教科書, 配付資料
- 自分のホームディレクトリ（ホームフォルダ）以下に保存されているファイル
- 小テストでは紙媒体のものは参照可能
- 上記以外の情報を参照することは不正行為とする  
例：USBで接続された機器に保存されているファイルの参照  
ネットワークを介した情報の参照、など