プログラミングII

http://bit.ly/Prog3i

計算モデル（2）

前期 第9週

2019/6/18

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

/* (2) 文字列の入力 */
printf("数字を入力してください。\n");
scanf("%s", input);
```

配列fin_statesに変更する
(終了状態が要素となった配列)

「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isaccpect()を作り、ここで呼び出す

まずは、先週のsample108.cにこれらの改良をしてみます

【課題9-1】

「sample108.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9-2】

「sample108.c」に、「current_stateが、配列fin_statesに含まれているかを判別する」関数isaccpect()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を戻す
- 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を戻す

【課題9-2】

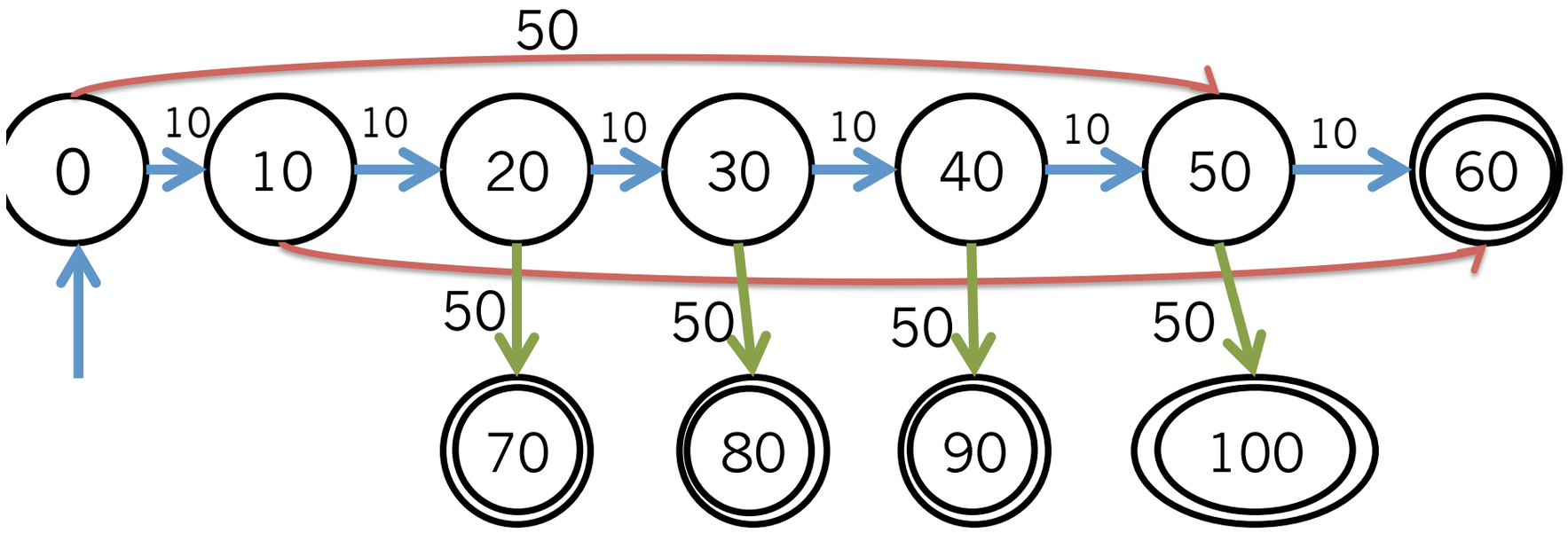
関数isaccept()を呼び出す箇所は以下のように変更します。

```
    /* (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8-1と同じく偶数の入力は受理されます

【課題9-3】

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, 80, 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

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）**

- **プログラムの提出はGitHubを使用する。**

小テストについて

小テスト中に参照できるもの

□ **教科書, 参考書, 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
Webブラウザ、ネットワークを介した情報の参照
自分のPCを使用する、など