

プログラミング応用
<http://bit.ly/kosen02>

Week9@後期
2016/11/24

- 12/8(木)は米国出張のため休講

本日の内容

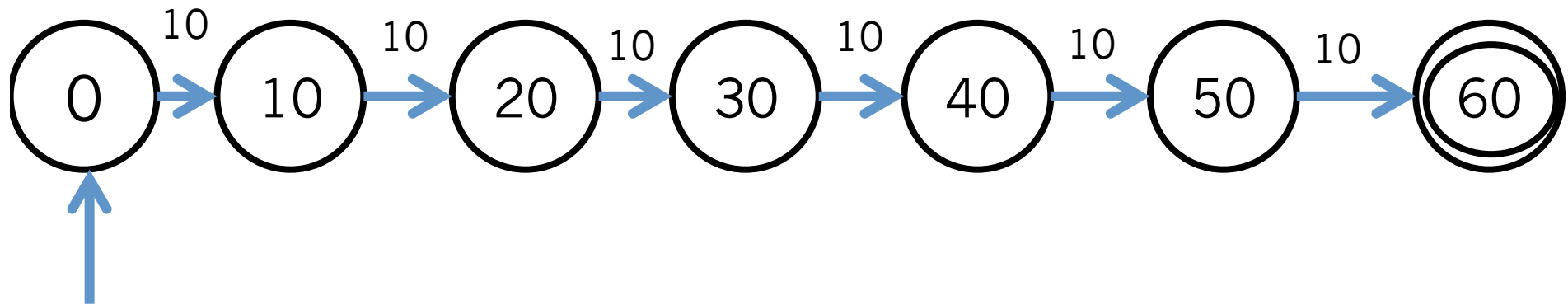
- 講義：
 - オートマトンと状態遷移図、正規表現
 - オートマトンのいろいろな例
 - C言語による実装方法
- 演習：
 - 状態遷移図からC言語プログラムを作る
 - 正規表現からC言語プログラムを作る

自動販売機の例

- オートマトンとはなにか？という問いの前に。。。
- 旧式の自動販売機を考える
 - 1本60円のジュースを売る
 - 10円玉もしくは50円玉のみを受け付ける
 - 60円に達したらジュースを出して終了

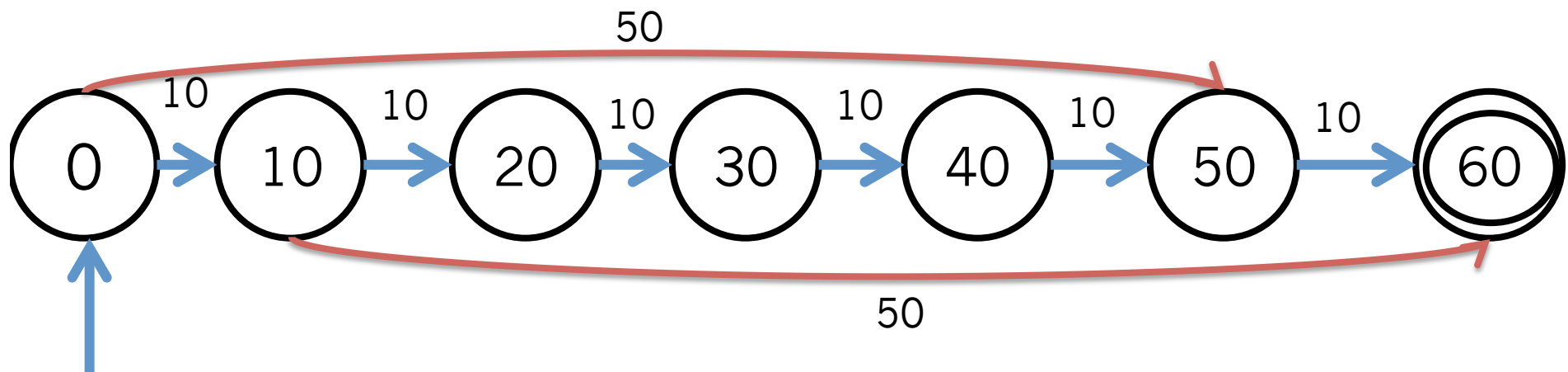
10円玉だけの場合。。。。

- 0からはじまり、10円玉を投入する度に次の状態に遷移していく



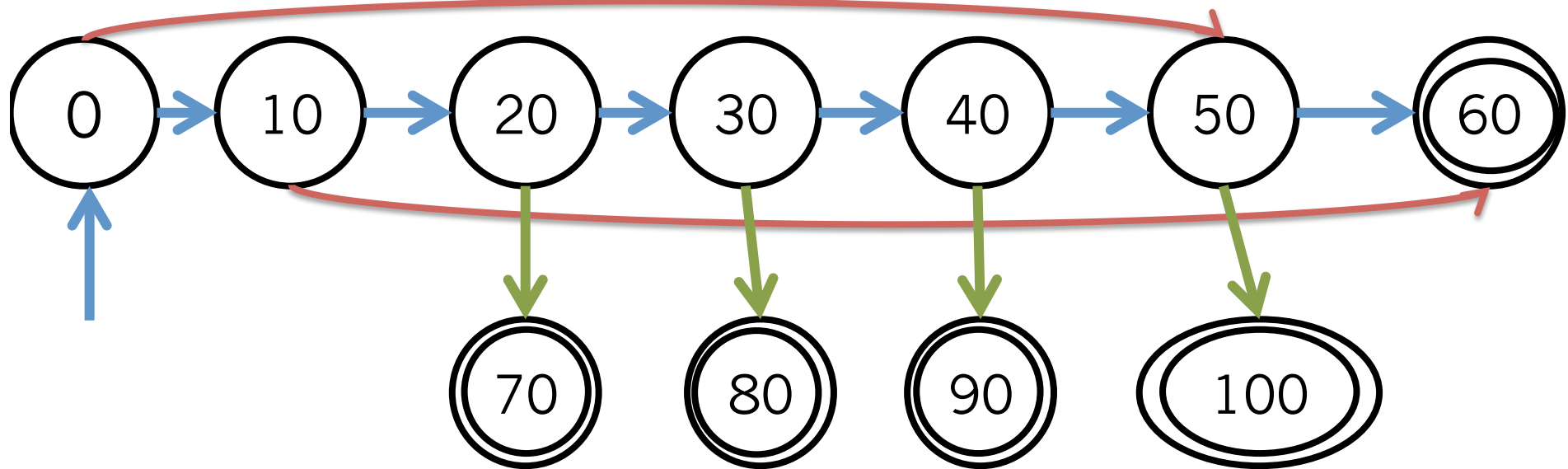
10円玉+50円玉だけの場合。。。。

- 0と10、50からの遷移を追加
これで十分？



10円玉+50円玉だけの場合。。。。

- 20, 30, 40, 50から50円を入れたときも
ジュースを出して終了

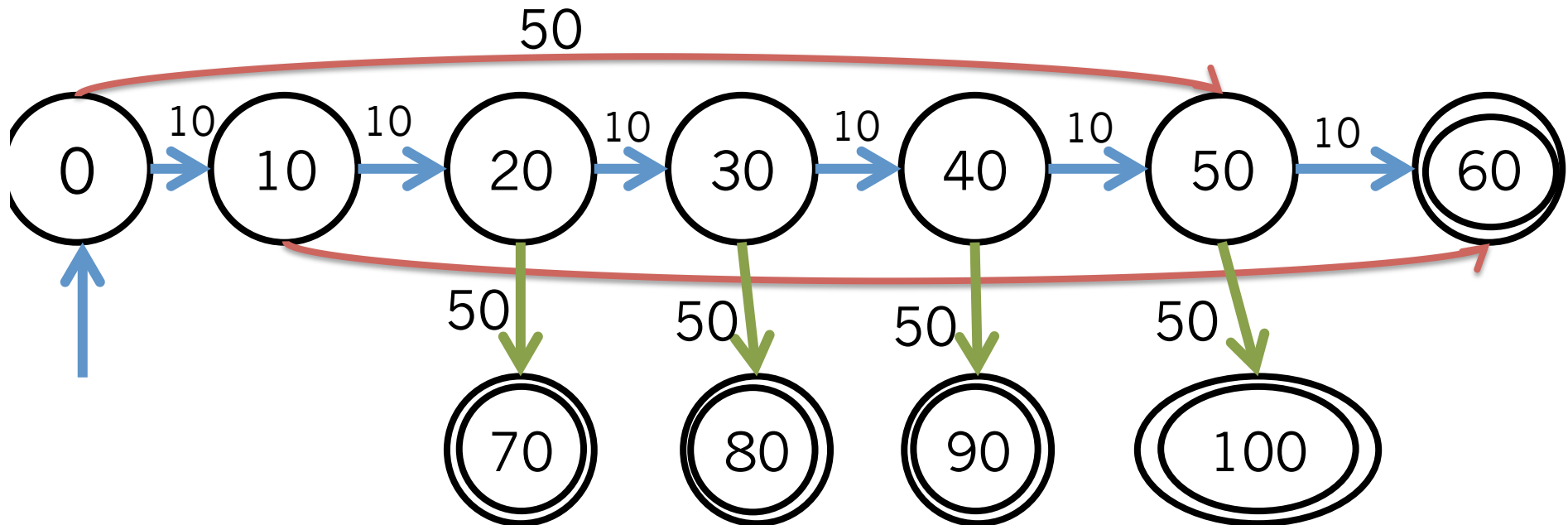


オートマトンとは？

- オートマトン
 - 状態と状態間の遷移を表現したモデル
 - 自販機の例ならば入っているお金の合計が状態、投入するコイン(10 or 50)が遷移
- 決定性有限オートマトン
 - 状態の数が有限個(有限)
 - 遷移先が一意に決まる(決定性)

オートマトンの「受理」

- 入力を先頭から読み込み、末尾まで到達したときに終了状態に居れば「**受理**」
入力が10円玉 + 50円玉 : 受理
入力が10円玉 + 10円玉 : 受理しない

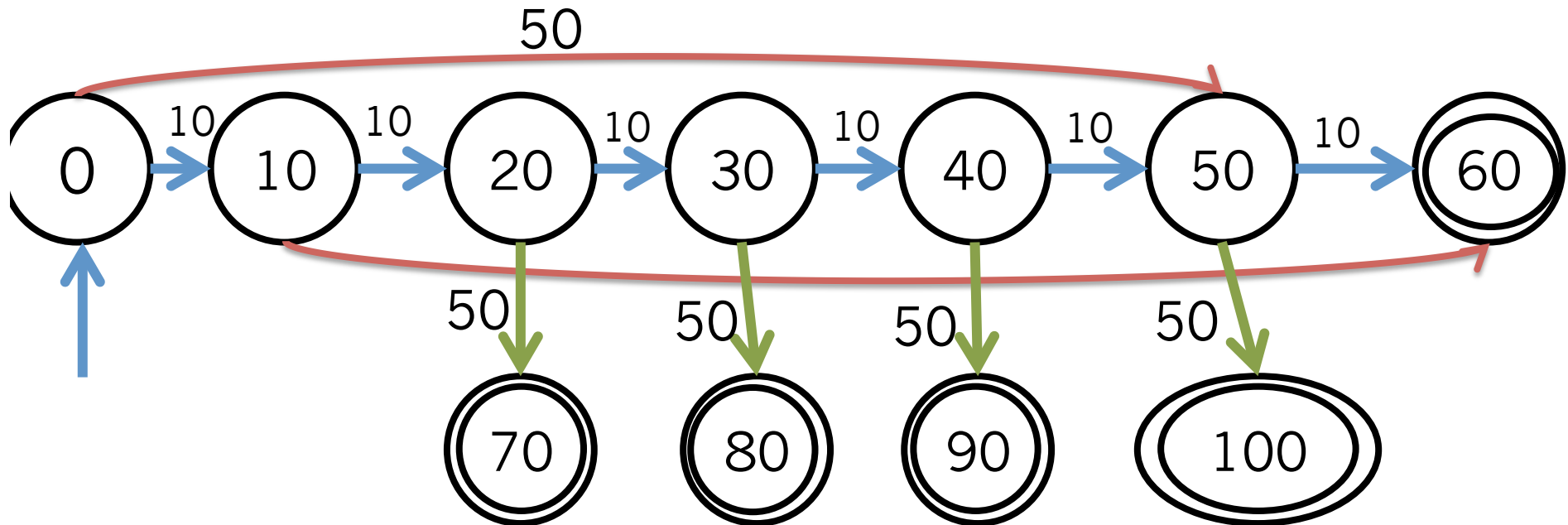


オートマトンの表現方法

1. 状態遷移図
2. 形式的定義
3. 正規表現

1. 状態遷移図でオートマトンを表現

- オートマトンの状態と遷移をグラフで表現
 - ノードは状態を表現
 - 初期状態には入り矢印をつける。終了状態は二重丸。
 - 辺：遷移先を表現



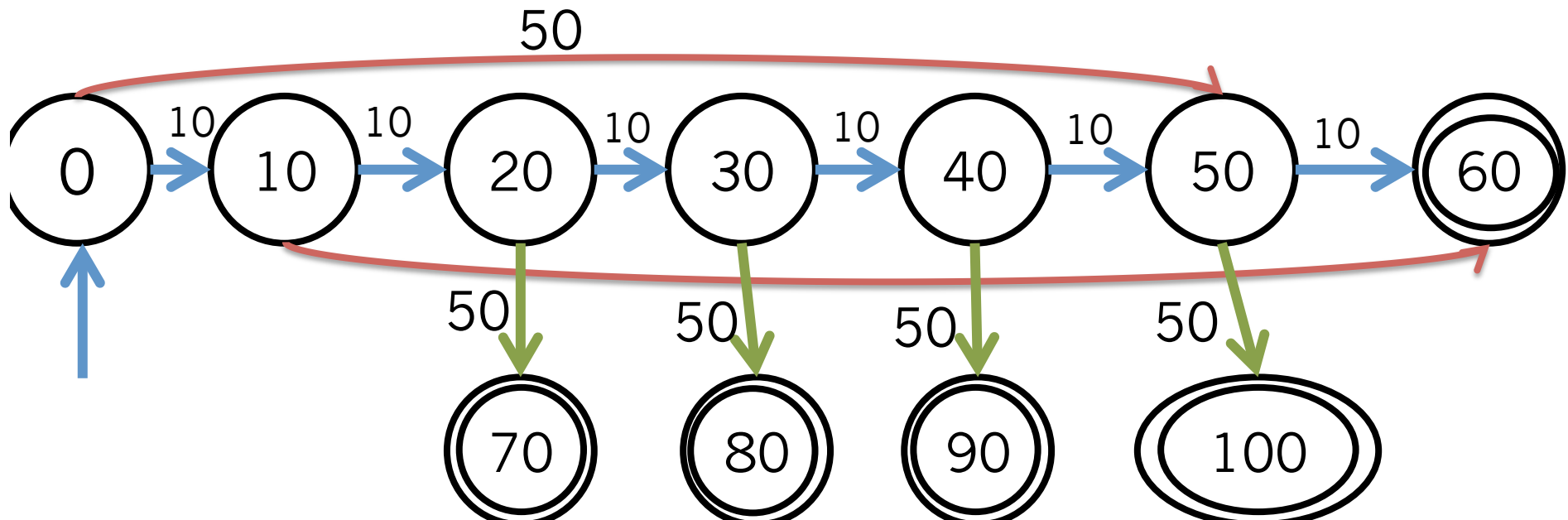
2. 形式的定義でオートマトンを表現

- 形式的定義の5要素
 1. K : (状態集合)
 2. Σ : アルファベット(入力として受け付ける記号)
 3. q_0 : 初期状態
 4. δ : 状態遷移
 5. F : 終了状態

自動販売機の形式的定義

例) 自動販売機の場合

1. $K = \{0, 10, 20, 30, 40, 50, 60, \dots, 100\}$
2. $\Sigma = \{10, 50\}$
3. $q_0 = 0$
4. $\delta = \{f(0, 10)=10, f(0, 50)=50, f(10, 10)=20, f(10, 50)=60, \dots, f(50, 50)=100\}$
5. $F = \{60, 70, 80, 90, 100\}$



3. 正規表現によるオートマトンの表現

- 正規表現を用いて記述することも出来る
 - 正規表現の例
(+は「または」、*は0回以上の連続を表す)

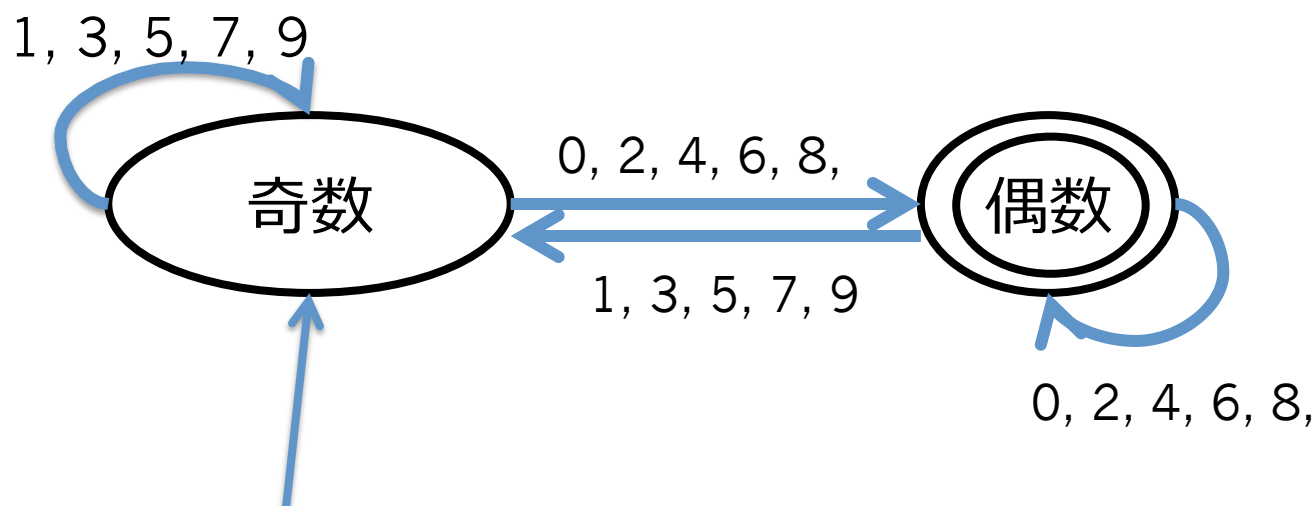
$(0+1)^*5$

この正規表現は「0もしくは1が0回以上連続し、最後が5で終わる数字列」を受理する

0005 : 受理
0105 : 受理
405 : 受理しない

例1-1) 偶数を受け取る状態遷移図

- 入力を先頭から読み込み末尾の数値が偶数ならば受理
 - 「124」を入力すると、
開始状態(奇数)→1(奇数のまま)→
2(偶数に遷移)→4 (偶数状態で終了=受理)

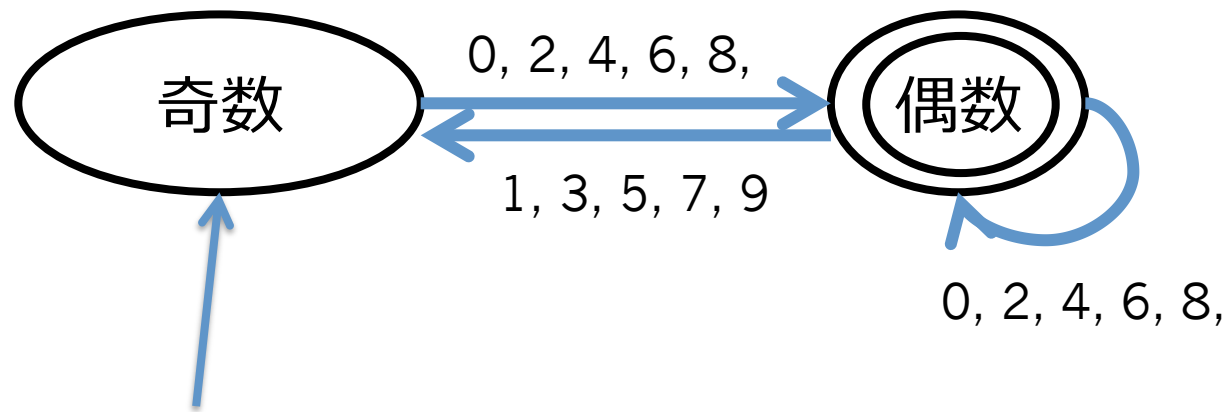


例1-2) 偶数を受理する正規表現

$(0+1+2+3+4+5+6+7+8+9)^*(0+2+4+6+8)$

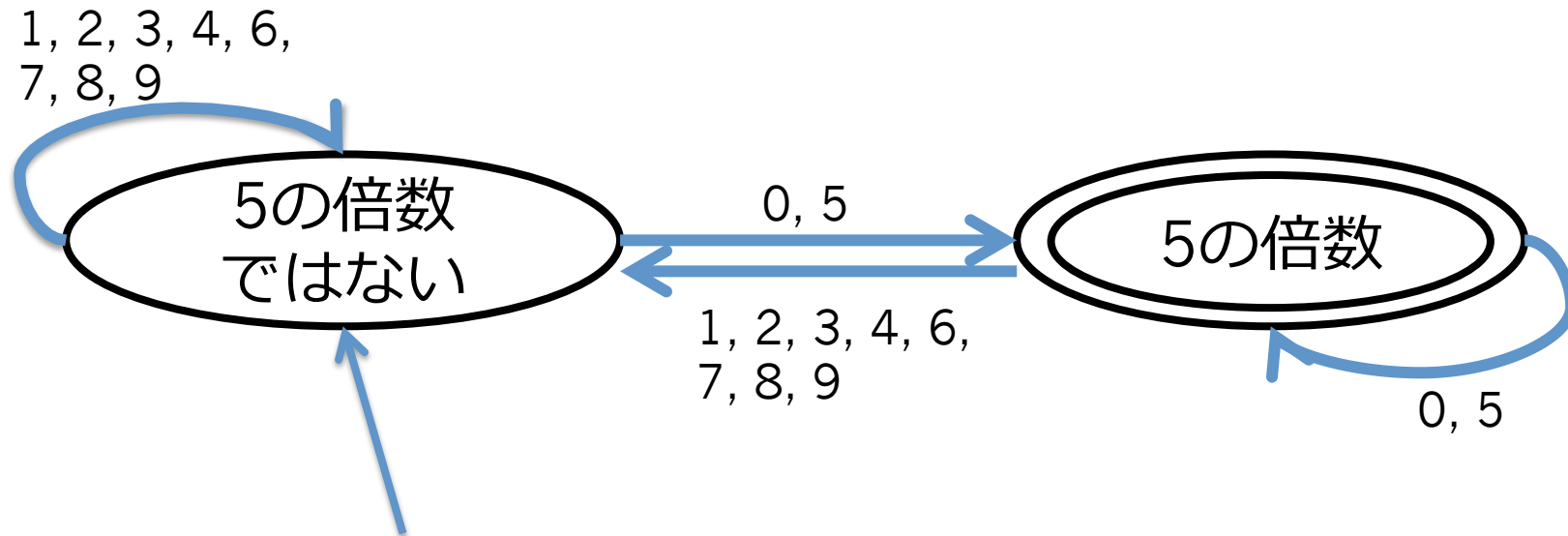


等価



例2-1) 5の倍数を受理する状態遷移図

- 5の倍数は末尾が0もしくは5
- 入力を先頭から読み込み末尾の数値が0, 5ならば受理



例2) 5の倍数を受理する正規表現

$(1+2+3+4+5+6+7+8+9)^*(0+5)$

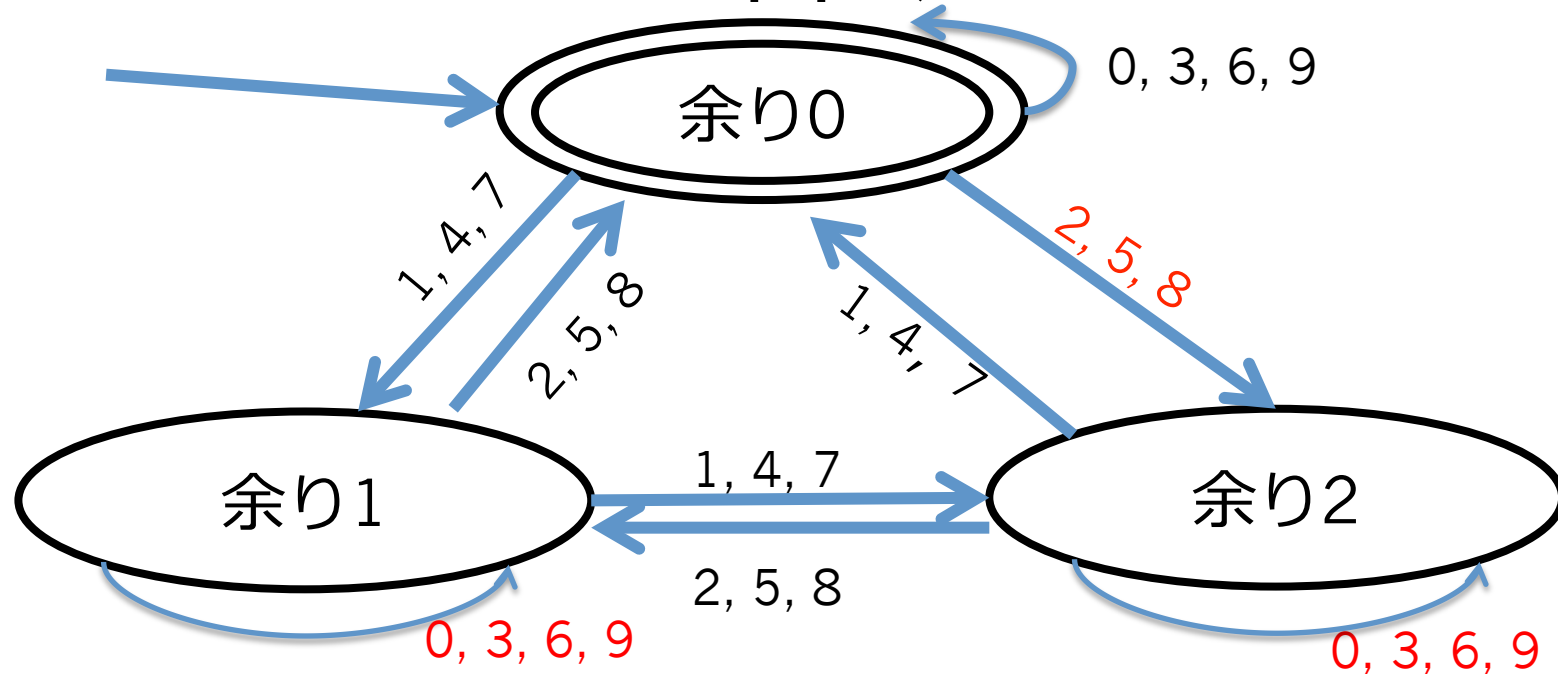
1から9までの数が0回以上連続して、最後は0か5で終わる

例3) 3の倍数を受理するオートマトン

- 3の倍数を受理するオートマトンは？
 - 末尾が3で割り切れるか判定すれば十分？
No! → 13は3の倍数ではない
- 3の倍数 = 各位の数字の和が3の倍数
 - 123: $1 + 2 + 3 = 6$ (3の倍数)
 - 124: $1 + 2 + 4 = 7$ (3の倍数ではない)

例3) 3の倍数を受理するオートマトン

※switch文のそれぞれのcaseにbreakを書く

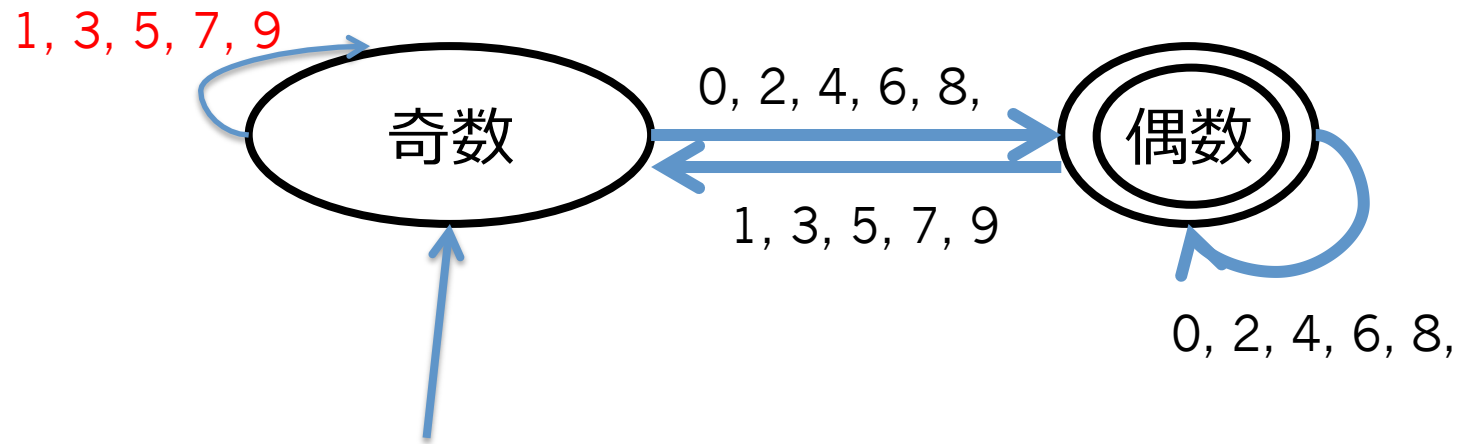


「123」を入力した場合：
1を3で割ると余り1なので余り1に遷移
(1+2)を3で割ると余り0なので余り0に遷移
(0+3)を3で割ると余り0なので遷移せず。
終了状態なので123は受理

本日の内容

- 講義：
 - オートマトンと状態遷移図、正規表現
 - オートマトンのいろいろな例
 - C言語による実装方法
- 演習：
 - 状態遷移図からC言語プログラムを作る
 - 正規表現からC言語プログラムを作る

C言語によるオートマトンの実装



必要な処理の流れ

1. 必要な変数を宣言

(ユーザからの入力を代入する変数、現在の状態を管理する変数)

2. 文字列を入力させる

3. while文で先頭から1文字ずつ読み込みながら switch文で状態を遷移させる

4. すべて読み込み終わったときに終了状態に居れば受理

1. 2.変数宣言

```
// 1. 必要な変数を定義
char input[100]; // 入力を読み込む
int i=0; // カウンタ変数
///// 状態を管理する変数(0: 奇数、1: 偶数)
int current_state=0; // 現在の状態
int fin_state=1; // 終了状態(1:偶数)
// 2. キーボードから文字列を読み込み
printf("数字を入力してください。\\n");
scanf("%s", input);
```

3. 状態遷移

```
16 // 3. 先頭から順に読み込みながら状態遷移
17 while(input[i]!='\0') {
```

while文の中に遷移する処理を書く

```
38 ;
39 i++;
40 }
```


3. 状態遷移

```
16 // 3. 先頭から順に読み込みながら状態遷移
17 while(input[i]!='\0') {
18     switch(current_state) {
19         case 0: // 現在の状態が0 (奇数) の場合
20             // 次の遷移先状態を代入
21             if(input[i]=='0' || input[i]=='2' ||
22                input[i]=='4' || input[i]=='6' ||
23                input[i]=='8') {
24                 current_state=0;
25             } else {
26                 current_state=1;
27             }
28         case 1: // 現在の状態が1 (偶数) の場合
29             // 次の遷移先状態を代入
30             if(input[i]=='1' || input[i]=='3' ||
31                input[i]=='5' || input[i]=='7' ||
32                input[i]=='9') {
33                 current_state = 0;
34             } else {
35                 current_state = 1;
36             }
37     }
38     printf("読み込んだ数値 : %c  遷移先 : %d\n", input[i], current_state);
39     i++;
40 }
```

修正 !

current_state = 1;

修正 !

current_state = 0;

4.終了状態にいるか判定

```
42    // 終了状態にいるか判定
43    if(current_state==fin_state) {
44        printf("受理する\n");
45    } else {
46        printf("受理しない\n");
47    }
48
49    return 0;
```

演習0

- 今回の演習では、状態遷移図、正規表現、C言語による実装を扱います。
- C言語プログラム以外は提出不要です

演習1

- 偶数を受理する状態遷移図、正規表現、C言語プログラムを、講義資料を見ながら書いて理解しなさい。

演習2

- 3の倍数を受理するC言語プログラムを講義資料の状態遷移図を参考に baisu3.c というファイル名で作成しなさい
- なお、current_stateやfin_stateに設定する値は以下のようにすること
 - 余り0の状態 : `current_state = 0;`
 - 余り1の状態 : `current_state = 1;`
 - 余り2の状態 : `current_state = 2;`

演習3

- 4の倍数を受理するオートマトンの状態遷移図を記述しなさい

※ヒント：余り0から余り3を状態にする

※C言語プログラムは長くなるので
時間がある人だけでok

※正規表現は長くなるので不要です

次回

- 次回は遷移先が一意に決まらない
非決定性オートマトンを扱います