基本情報技術Ⅰ

5. オートマトンと形式言語

菊池浩明

講義目標

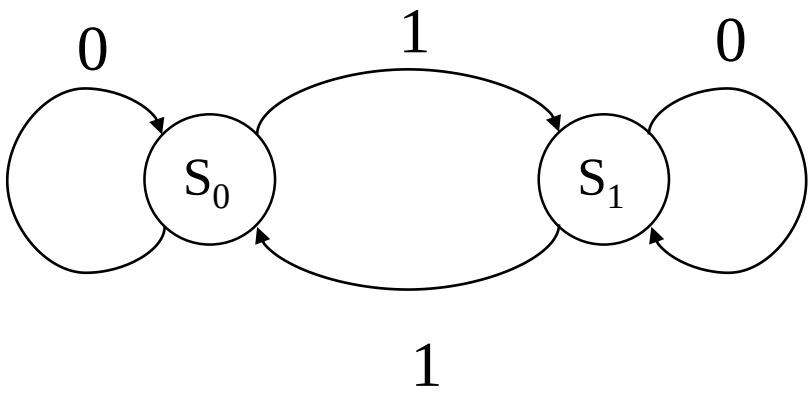
- 教科書
  - 1-4 情報・通信に関する理論
    - » 3. オートマトン
    - » 4. 正規表現
    - » 5. 形式言語
    - » 6. 逆ポーランド

宿題（次回小テスト範囲）

- 1章練習問題
  - （オートマトンに関する問題がない．過去の小テストなどを復習せよ）

1. オートマトン ★頻出

- 有限オートマトン（finite state machine)
  - 有限の「状態=メモリ」を持つシステム
  - 参考) stateless（状態を持たない）
- 状態遷移図
  - 状態の集合K={S_0,S_1}
  - 入力記号集合Σ={0,1}
  - 状態遷移δ: KxΣ→K

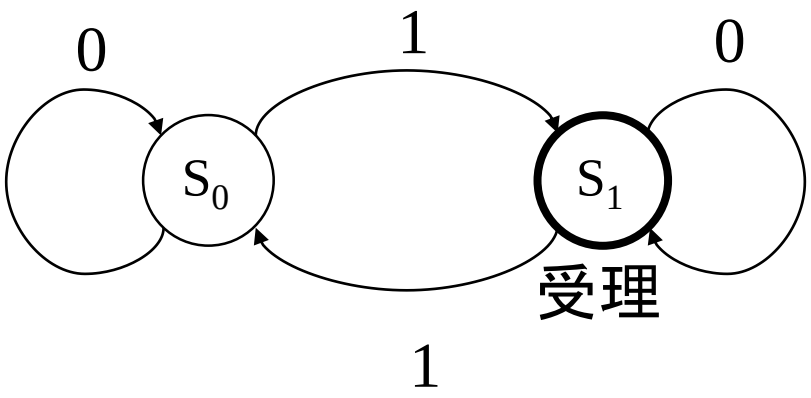


状態遷移関数

- 状態の遷移を決める
  - δ: KxΣ→K
  - δ(現在の状態，入力)=次の状態
  - $\delta(S_0,1)=S_1$, S_0=「奇数」, S_1=「偶数」
  - 例) 初期状態 S_0, 入力列 1011 の時の最終状態を求めよ.
    - » $S_0 \rightarrow S_1 \rightarrow S_1 \rightarrow S_0 \rightarrow \underline{S_1}$

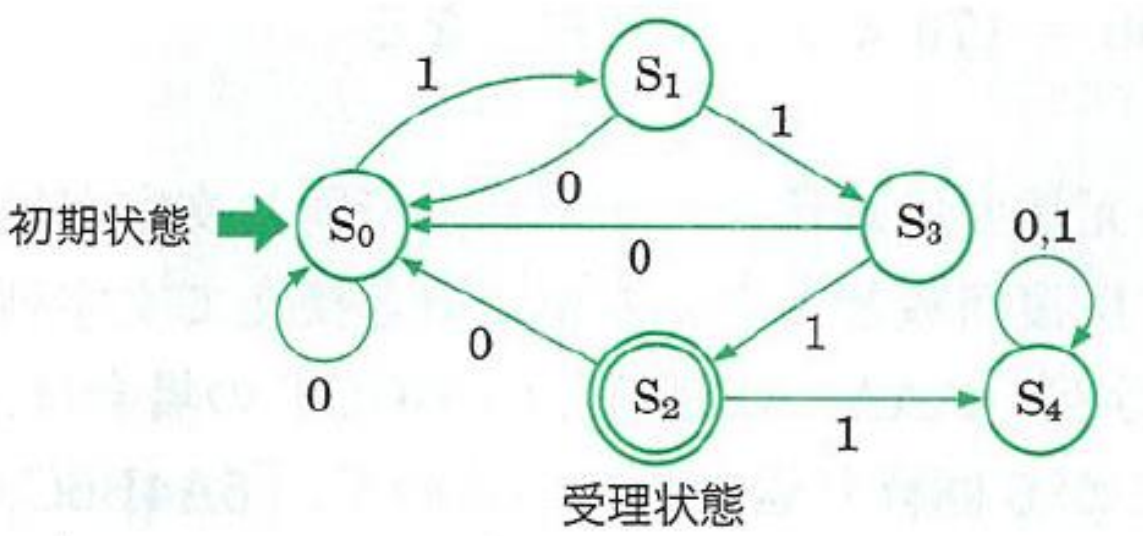
入力\　現在の状態	S0	S1
0	S0	S1
1	S1	S0

状態遷移表



例1.

- 次で定義される有限オートマトンで受理される入力文字列はどれか.
  - ア) 01011
  - イ) 01111
  - ウ) 10111
  - エ）11110



例2

- 次の状態遷移図で受理されない(最終状態がE)文字列はどれか？
  - ア +0010　イ -1　ウ 12.2　エ 9._

現状態＼入力	空白	数値	符号	小数点	他
A	A	B	C	D	E
B	A	B	E	D	E
C	E	B	E	D	E
D	A	E	E	E	E

2. 正規表現

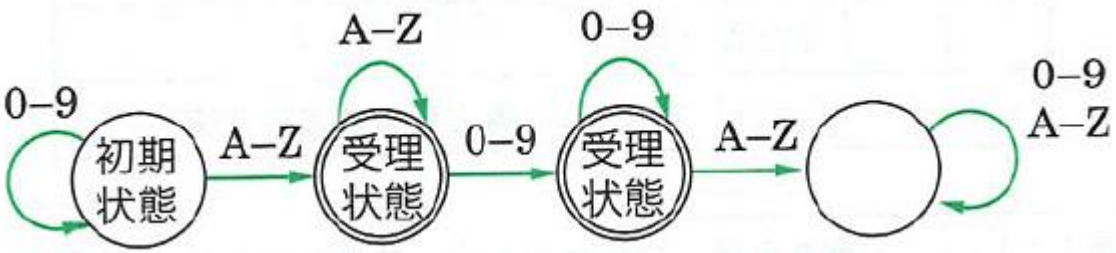
- 正規表現 (regular expression)
  - 演算子で定められる記号列の集合
  - 例）dir FIT*.pptx (FITで始まるPPTファイルを表示する), grep, sedコマンドなど
- メタ文字
  - [m-n]:　mからnまでの連続した文字
  - * :　直前の正規表現の0回以上の繰り返し
  - + : 直前の正規表現の1回以上の繰り返し

(参考)実際に使われる正規表現例

- 日付
  - 正規表現: ¥d+/¥d+/¥d+
  - 例）2014/11/11, 99/1/3, (2014/13/40 も受理）
- URL
  - https?://[¥w¥d/%#$&?()~_.=+-]+
  - 例) http://www.meiji.jp
    https://google.co.jp/?gws_rd=ssl#q=%E6%98%8E%E6%B2%BB (「明治」で検索）

例3)

- 正規表現 [A-Z]+[0-9]*で受理されない文字列はどれか？
  - ア ABC12 イ AB3　ウ 123 エ ABC



4. 形式言語

- BNF (Backus-Naur Form)
  - 形式言語の構文規則を定める表記法
  - トークン(文字列要素）
  - 終端記号･非終端記号 (<AAA>)
- 構文記述記号
  - ::=　　定義
  - |　　または
  - [ ]　　省略可能要素

例4

- BNF <S>::= 01| 0<S>1 で定義されるビット列はどれか？

  ロア 010010 イ 010101 ウ 011111 エ 000111

例5

- 次のBNFで定義される<数値>はどれか

  <数値>::= <数字列>|<数字列>E<数字列>
  | <数字列>E<符号><数字列>

  <数字列>::= <数字>|<数字列><数字>

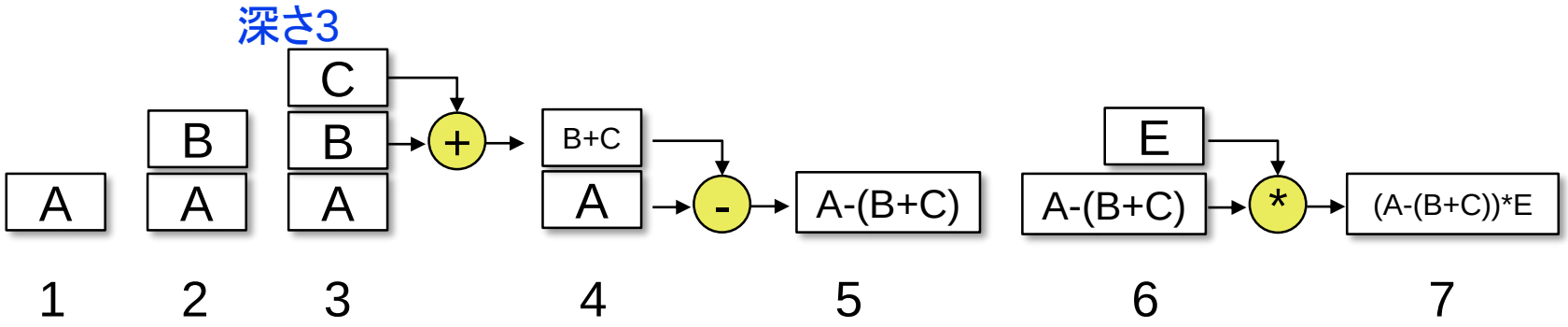
  <数字>::= 0|1|2|3|4|5|6|7|8|9

  <符号>::= +|-

  ❑ア -12, イ　12E-3, ウ +12E-3, エ +12E10

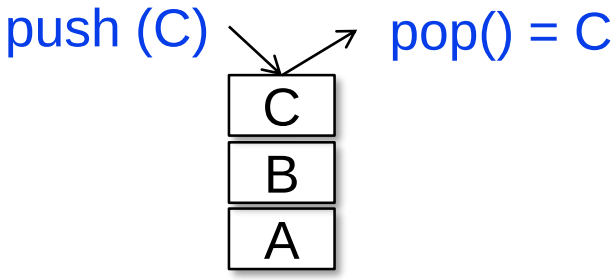
3. 逆ポーランド表記法 ★★頻出

- 後置表記法
  - 演算式を表す表記法の一つ．スタック（先入れ後出しデータ構造）を用いて，括弧を省略する．
  - 1. オペランド(変数や値）はスタックにpush（積む）
  - 2. 演算子(+, *, =)はスタックから要素を二つpop(下す)して，演算を行い，結果をpushする．
  - 例) ABC+-E* は E*(A-(C+B)) と同値



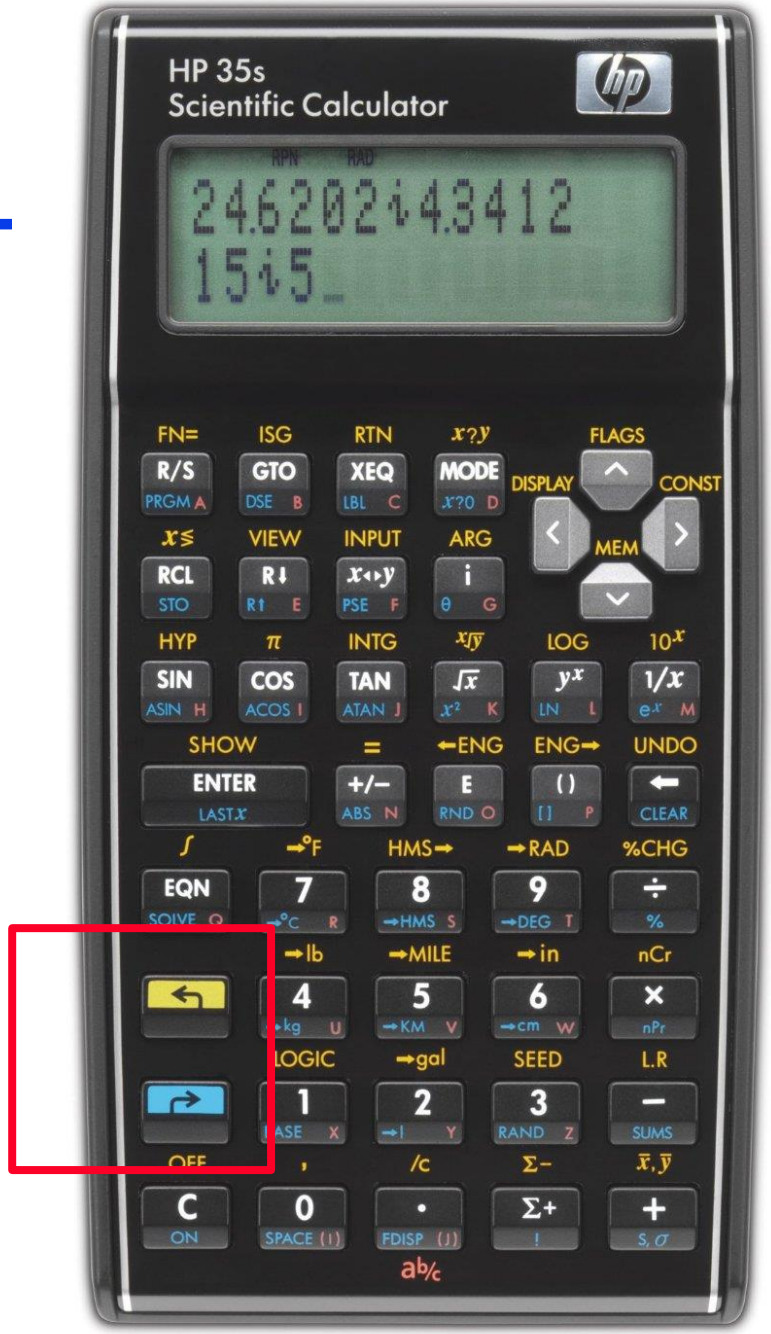
スタック

- Stack
  - push(x)　　xを格納
  - pop()　　データ取出し
  - 後入れ先出し
    Last-In First-Out (LIFO)



逆ポーランド電卓

- HP 35s
  - ハイエンド関数電卓
  - 6,990円(amazon)



例6

- A=1, B=3, C=5, D=4, E=2の時，式AB+CDE/-+の演算結果を求めよ.

例7

- スタックの深さが最も少ない式はどれか？
  - ア ab+c+d+　　((a+b)+c)+d
  - イ ab+cd++　　(a+b)+(c+d)
  - ウ abc++d+
  - エ abc+d++

まとめ

- 有限の状態に基づいて受理を判断するシステムを有限（　　）という．状態遷移図または（　　）で定義をする．
- メタ文字で記述される入力文字列集合の表記法を（　　）という．
- 逆ポーランド記法では，演算子が後から来るので（　　）とも呼ばれる．オペランドを（　　）にpushし，演算子は（　　）した二つの値を計算して結果をpushする．