

ソフトウェアモデル論(2013年度) 第2回・2013/10/03

桑原 寛明
情報理工学部 情報システム学科

集合

(復習)

- 「もの」(要素)の集まり
 - 同じ要素は高々1つ
- 部分集合、べき集合
- 集合上の演算
 - 和、積、差、補集合
- 直積
 - 順序対

ソフトウェアモデル論(2013/10/03)

2

関係

(復習)

- 「もの」と「もの」の関係
- 直積の部分集合
 - 関係のある「もの」の組合せを全部集める
- 反射、対称、反対称、推移

ソフトウェアモデル論(2013/10/03)

3

順序関係

- 反射的かつ反対称的かつ推移的な関係
- 全順序関係
 - 集合 S 上の順序関係 R
 - S の任意の2つの要素間に R または R^{-1} が成立

ソフトウェアモデル論(2013/10/03)

4

同値関係

- 反射的かつ対称的かつ推移的な関係

ソフトウェアモデル論(2013/10/03)

5

関係の合成

- 2つの関係を順にたどって作る新しい関係
 - $R_1 \subseteq S \times T, R_2 \subseteq T \times U$
 - $R_1 \circ R_2 = \{(s,u) \mid \exists t \in T. sR_1t \text{ かつ } tR_2u\}$
- $\{(1,a)\}$ と $\{(a,A)\}$ を合成すると $\{(1,A)\}$

ソフトウェアモデル論(2013/10/03)

6

反射推移閉包

- 集合 S 上の関係 R の反射推移閉包
 - R を含み反射的かつ推移的な最小の関係
 - R を0回以上たどって到達できる要素間の関係
- $S = \{ a, b, c \}$, $R = \{ (a,b), (b,c) \}$ とすると
 $R^* = \{ (a,a), (a,b), (a,c), (b,b), (b,c), (c,c) \}$

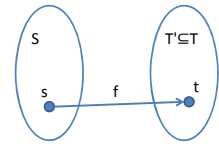
ソフトウェアモデル論(2013/10/03)

7

関数(写像)

- 集合 S の各要素に集合 T の要素を1つだけ対応付ける
 - $f: S \rightarrow T$
 - $f(s) = t$

- 定義域
 - 対応付けの元の集合
- 値域
 - 対応付けの先の集合

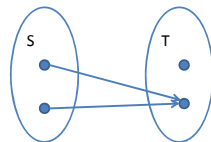
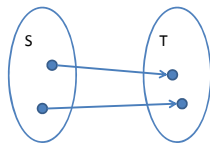


ソフトウェアモデル論(2013/10/03)

8

関数(写像)

- 単射
 - 同じ要素に対応付けない
- 全射
 - 全要素が値域に含まれる
- 全単射
 - 単射かつ全射

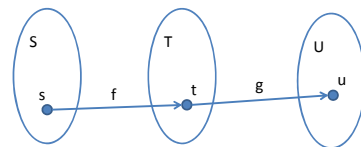


ソフトウェアモデル論(2013/10/03)

9

関数の合成

- $f: S \rightarrow T$ と $g: T \rightarrow U$ を合成
 - $h = g \circ f: S \rightarrow U$



ソフトウェアモデル論(2013/10/03)

10

有限オートマトンと正規表現

ソフトウェアモデル論(2013/10/03)

11

順序機械

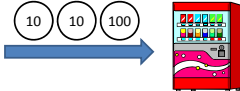
- ある入力を与えると、内部で何かを行って、何かを出力する機械
 - 次の出力は、今までの入力と次の入力によって決定される
- 今までの入力によって機械の内部状態が決定される
 - 入力による内部状態の変化が状態遷移

ソフトウェアモデル論(2013/10/03)

12

例(自動販売機)

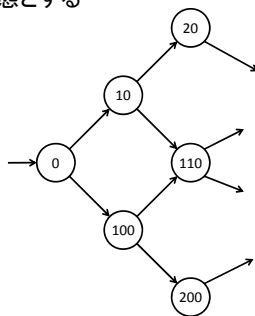
- 120円のジュースを買いたい
 - 100円、10円、10円の順に硬貨を投入する
 - 10円、10円、100円の順に硬貨を投入する
- 自動販売機は硬貨がいくら投入されたか記憶しなければならない
- 120円投入されたらジュースのランプを点灯する



ソフトウェアモデル論(2013/10/03) 13

例(自動販売機)

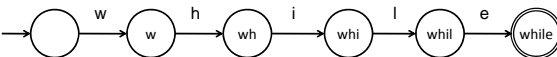
投入された金額を状態とする
状態遷移図



ソフトウェアモデル論(2013/10/03) 14

その他の例

- 順序回路
- コンパイラの字句解析・構文解析

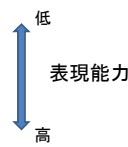


ソフトウェアモデル論(2013/10/03) 15

オートマトン

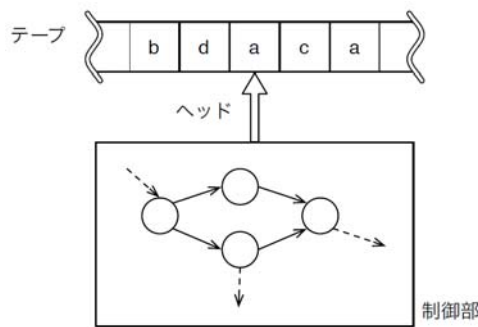
- 順序機械を一般的に記述するモデル
 - どのような順序機械でも表現できる枠組み
- 計算機が実行する「計算」を抽象的に表現するモデル

- 有限オートマトン
- プッシュダウンオートマトン
- 線形有界(拘束)オートマトン
- チューリング機械



ソフトウェアモデル論(2013/10/03) 16

オートマトンの概念図



ソフトウェアモデル論(2013/10/03) 17

有限オートマトン

- 以下の制限を加えたオートマトン
 - 読み出しのみ
 - 1マス読んだら必ずヘッドを右へ1マス移動
 - 左端のマスから開始

ソフトウェアモデル論(2013/10/03) 18

有限オートマトンの動作

1. ヘッドの位置のマスを読む
2. 読んだ記号に従って状態遷移する
3. ヘッドを1マス右へ移動する
4. テープ上のすべての記号を読んで状態遷移が終わった時に、状態が最終状態になっていれば入力された記号列を受理する

ソフトウェアモデル論(2013/10/03)

19

決定性有限オートマトン (DFA)

$$M = (Q, \Sigma, \delta, q_0, F)$$

Q : 状態の有限集合 ($\neq \emptyset$)

Σ : 入力記号の有限集合

δ : 状態遷移関数 ($Q \times \Sigma \rightarrow Q$)

q_0 : 初期状態 ($\in Q$)

F : 受理状態の集合 ($\subseteq Q$)

ソフトウェアモデル論(2013/10/03)

20

DFAの例

- ab の繰返しを受理するDFA

$$Q : \{S_0, S_1, S_2\}$$

$$\Sigma : \{a, b\}$$

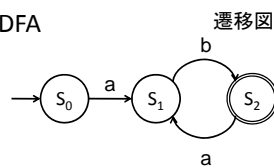
$$\delta : (S_0, a) \rightarrow S_1,$$

$$(S_1, b) \rightarrow S_2,$$

$$(S_2, a) \rightarrow S_1$$

$$q_0 : S_0$$

$$F : \{S_2\}$$



遷移表

	a	b
S_0	S_1	-
S_1	-	S_2
S_2	S_1	-

ソフトウェアモデル論(2013/10/03)

21

主要な概念

- アルファベット
 - 記号の有限集合
- 語
 - 記号列
- アルファベットから作られる語の集合
 - $\Sigma = \{0, 1\}$ ならば
 - $\Sigma^* = \{\epsilon, 0, 1, 00, 01, 10, 11, 000, \dots\}$
- Σ 上の言語L
 - $L \subseteq \Sigma^*$

ソフトウェアモデル論(2013/10/03)

22

遷移関数の拡張

- 状態と語を受け取って状態を返すように拡張

$$\hat{\delta} : Q \times \Sigma^* \rightarrow Q$$

$$\hat{\delta}(q, \epsilon) = q$$

$$\hat{\delta}(q, wa) = \delta(\hat{\delta}(q, w), a)$$

ソフトウェアモデル論(2013/10/03)

23

受理言語

- $\delta(q_0, w) \in F$ ならば語wは受理される
 - 開始状態から開始して受理状態で終了
- オートマトンMの受理言語

$$L(M) = \{w \mid w \in \Sigma^* \wedge \delta(q_0, w) \in F\}$$
 - 受理される語の集合
- $L(M_1) = L(M_2)$ の時 M_1 と M_2 は等しい
 - オートマトンが等しいとは受理言語が等しいこと

ソフトウェアモデル論(2013/10/03)

24