オートマトンと言語

大阪分散技術コミュニティ

2019 年 2 月 2 日

タイトル　オートマトンと言語
著者　Michael Sipser
訳者　太田和夫, 田中圭介
出版日　2008/5/21
出版社　共立出版
ISBN10　4320122070
ISBN13　978-4320122079
ページ数　240
言語　ja
内容　MIT 屈指の名講義の講義ノートをまとめた書

1　Notation

使用する記号と用語についてまとめる。

Σ: Alphabet　空でない有限集合
s: Symbol(文字)　アルファベットの元
ω: String over an alphabet　有限の文字列
$|\omega|$: Length　文字列の長さ
ε: Empty string(空列)　$|\varepsilon| := 0$
L: Language(言語)　文字列の集合
$\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$
2^A:　A のべき集合
$\mathbb{N}$:　0 を含む自然数

$w = s_1 s_2 \cdots s_n$ であり、文字列と文字は区別される。$|w| = n$ である。

star(スター演算)

集合 A に対してスター演算を以下で定義する。ただし $|s| = 0$ なる元は ε に限る。

$$A^* = \{x_1 x_2 \cdots x_k | x_i \in A, k \in \mathbb{N}\}$$

例　$\Sigma = \{0, 1\}$ とすると、$\Sigma^* = \{\varepsilon, 0, 1, 00, 01, 10, 11, \cdots\}$
この例から分かるように、$s \in \Sigma, w \in \Sigma^*$ である。

2 Automaton

有限オートマトン (finite automaton)M を以下で定義する。

$$M = < Q, \Sigma, \delta, q_0, F >$$

Q: States(状態集合)　空でない有限集合
Σ: Alphabet　空でない有限集合
δ: Transition functions(遷移関数)　$Q \times \Sigma \to Q$
q_0: Start state(開始状態)　$q_0 \in Q$
F: Set of accept states(受理状態集合)　$F \subset Q$

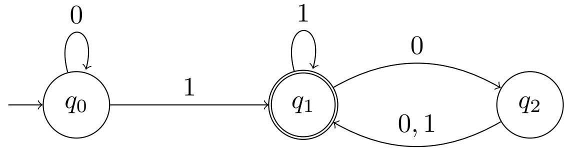
Q の元 q を状態 (state) と言う。オートマトンは初期状態 q_0 から遷移関数に従って動作する。ある入力 $\omega = s_0 s_1 \cdots s_n \in \Sigma^*$ に対して、

$$M(\omega) := \delta(\delta(\cdots \delta(\delta(\delta(q_0, s_0), s_1), s_2) \cdots, s_{n-1}), s_n)$$

を定義する。$M(\omega) \in F$ であれば M は入力 ω を受理 (accept) すると言う。受理しない時、M は入力 ω を拒否 (reject) すると言う。

3 State Diagram

オートマトンは状態遷移図 (state diagram) と呼ばれる図によって記述することができる。初期状態を矢印で表し、状態は丸で囲む。受理状態は二重丸で表す。



この図から、

$$M =< \{q_0, q_1, q_2\}, \{0, 1\}, \delta, q_0, \{q_1\} >$$

が分かる。ただしここで、

$$\delta(q_0, 0) = q_0, \delta(q_0, 1) = q_1$$
$$\delta(q_1, 0) = q_2, \delta(q_1, 1) = q_1$$
$$\delta(q_2, 0) = q_1, \delta(q_2, 1) = q_1$$

である。

オートマトンが正しく定義されるためには、任意の状態において、全ての文字に対する動作が決まらなければばならない。図において全ての状態から 0 と 1 の矢印が出ていることに注意されたい。

4 Regular Language

機械 M の言語 (language of machine M) とは、

$$M(L) := \{w \in \Sigma^* | M(w) \in F\}$$

で定義される集合である。ある言語 $L = \{w_1, w_2, w_3, \cdots\}$ が、

$$L = M(L)$$

を満たすならば、M は L を認識 (recognize) すると言う。

言語 L を認識するオートマトン M が存在する時、L を正規言語 (regular language) と呼ぶ。

5 Nondeterministic Automaton

先述の定義は決定性有限オートマトン (Deterministic Finite Automaton, DFA) と呼ばれ、遷移関数 $\delta : \Sigma \times Q \to Q$ を $\delta : \Sigma \times Q \to 2^Q$ に置き換えたものを、非決定性有限オートマトン (Nondeterministic Finite Automaton, NFA) と呼ぶ。もはや、文字を読み込んだ後の状態は一意に決まる必要はない。

6　Turing Machine

チューリング機械 (turing machine)TM を以下で定義する。

$$TM =< Q, \Sigma, \Gamma, \delta, q_0, q_{accept}, q_{reject} >$$

Q: States(状態集合)　空でない有限集合
Σ: Alphabet　空でない有限集合
Γ: Tape Alphabet　$\Sigma \subset \Gamma$, $\sqcup \in \Gamma$
δ: Transition functions(遷移関数)　$Q' \times \Gamma \to Q \times \Gamma \times \{L, R\}$
q_0: Start state(開始状態)　$q_0 \in Q$
q_{accpet}: Start state(開始状態)　$q_{accept} \in Q$
q_{reject}: Start state(開始状態)　$q_{reject} \in Q(\neq q_{accept})$

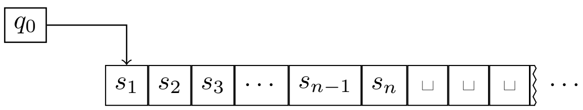
Q' とは Q から q_{accept} と q_{reject} を除いた集合である。
テープアルファベットとは、アルファベットに制御文字 (control symbol)

$$s' \notin \Sigma \wedge s' \in \Gamma$$

を足した集合で、必ず空白文字 (blank symbol)$\sqcup$ を含む。

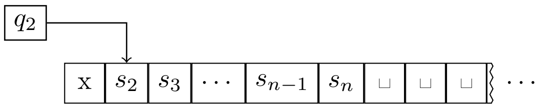
7　Diagram

チューリング機械は入力と、空白文字で埋められた無限長のテープを持ち、q_{accept} か q_{reject} のいずれかの状態になると動作を停止する。入力が $\omega = s_1 s_2 \cdots s_n \in \Sigma^*$ である時、チューリング機械はテープとヘッド (制御部) を持った以下のような図式で表現される。



初期状態 q_0 において、ヘッドは左端に位置する。ここから、遷移関数に従ってテープの内容を書き換えながら、ヘッドが左右に動いていく。このような描像をチューリング機械の計算と呼び、遷移関数の結果を参照する毎に、計算のステップが進むと考える。

例えば、$\Gamma = \Sigma \cup \{\mathrm{x}, \sqcup\}, \delta(q_0, s_1) = (q_2, \mathrm{x}, R)$ であれば、計算が 1 ステップ進むと



となる。ここでテープに空白でない制御文字が含まれていることに注意しよう。

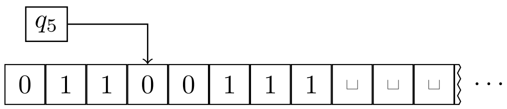
8　Configuration

チューリング機械の計算状況 (configuration) を以下のような列で表現する。

$$c_i = uq_j s_k v$$

ここで q_j は現在の状態を表わし、s_k は次にチューリング機械が読み込む文字である。u と v はヘッドより左側と右側の文字列を代表する。ただし、右側の空白文字は含めない。計算のステップが一つ進む時、計算状況 c_i は c_{i+1} に移る (yield) と言う。

例



計算状況は次の列で表現される。

$$011q_5 00111$$

チューリング機械が計算をする時、計算状況を並べることで列 C を得る。

$$C = c_0, c_1, \cdots, c_n, \cdots$$

チューリング機械の状態が $q_0, q_{accept}, q_{reject}$ のいずれかである時、それぞれの計算状況を

- 開始状況 (start configuration) c_0
- 受理状況 (accepting configuration) c_{accept}
- 拒否状況 (rejecting configuration) c_{reject}

と呼んで特に区別する。

受理状況と拒否状況をあわせて停止状況 (halting configuration) と呼ぶ。

9 Computation

ある入力 $\omega = s_0 s_1 \cdots s_n \in \Sigma^*$ に対して、$TM(\omega)$ を以下で定める。

$$TM(\omega) := \begin{cases} q_{accept} & \text{列 } C = c_0, \cdots, c_{accpet} \text{ が存在する。} \\ q_{reject} & \text{列 } C = c_0, \cdots, c_{reject} \text{ が存在する。} \end{cases}$$

一度でも停止状況になればチューリング機械は動作を停止するので、c_{accept} や c_{reject} から先の計算は考えない。このような有限の列が得られない時、チューリング機械はループ (loop) するという。この時 $TM(\omega)$ は定義されないとする。

10 Turing-recognizable

チューリング機械 TM の言語 (language of machine M) とは、

$$TM(L) := \{w \in \Sigma^* | TM(w) = q_{accpet}\}$$

で定義される集合である。ある言語 $L = \{w_1, w_2, w_3, \cdots\}$ が、

$$L = TM(L)$$

を満たすならば、TM は L を認識 (recognize) すると言う。

言語 L を認識するチューリング機械 TM が存在する時、L をチューリング認識可能 (turing-recognizable) と言う。

11 Turing-decidable

チューリング機械 TM が任意の $\omega \in \Sigma^*$ に対して、$TM(\omega)$ を定義する時、TM を判定装置 (decider) と言う。

言語 L が判定可能 (turing-decidable) とは、L が認識可能であって、判定装置であるような TM が存在することである。

12 Annotation

p15,—をつなげてるとき、その有向グラフを強連結 (strongly connected) という.

正確な定義は任意の 2 点間に有向路 (directed path) が存在することである。例えば図 0.16 だと頂点は繋がっているが (connected)、3 から 6 は辿ることができない。よって強連結

(strongly connected) とは言えない。

13 Questions

p30, 演習

0.1

a). 奇数

b). 負を含む偶数

c). 偶数

d). 偶数かつ奇数

e). $\{(0,0),(0,1),(1,0),(1,1)\}$

f). $\emptyset$

0.2

a). $\{1,10,100\}$

b). $\{m \in \mathcal{Z} | m > 5\}$

c). $\{n \in \mathcal{N} | n < 5\}$

d). $\{abc\}$

e). $\{\epsilon\}$

f). $\emptyset$

0.3

a). はい。

b). いいえ。

c). $\{x,y,z\}$

d). $\{x,y\}$

e). $\{(x,x),(x,y),(y,x),(y,y),(z,x),(z,y)\}$

f). $\{\{x,y\},\{x\},\{y\},\emptyset\}$

f). 集合 B の冪集合 (power set) は 2^B という記号で表すことが多い。

0.4

$a \times b$

0.5

2^c

0.6

a). 7
b). X, Y
c). 6
d). $X \times X, Y$
e). 8

0.7

a). 例えば、$a = a'$ or $b = b'$ によって関係 R を定めると
b). b
c). b mathématicha

0.8

pandax 宿題

0.9

pandax 宿題