

ソフトウェアモデル論(2011年度) 第6回・2011/11/04

桑原 寛明
情報理工学部 情報システム学科

正規言語

(復習)

- 正規文法が生成する言語
- 有限オートマトンが受理できる言語
- 正規表現で表現できる言語

ある言語が正規言語か否か判定するにはどうすればよいか

ソフトウェアモデル論(2011/11/04)

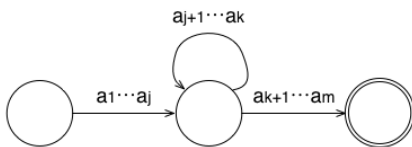
2

有限オートマトンが長い語を受理する場合

(復習)

- 初期状態から受理状態に到達する間にループが存在する
 - 同じ状態を2度以上通過する
- 状態数が n で語の長さが n 以上だと...

部屋割り原理
鳩ノ巣原理



ソフトウェアモデル論(2011/11/04)

3

反復補題

(復習)

- 正規言語 L に対して n が存在して、 $|z| \geq n$ なる任意の $z \in L$ について以下を満たすように z を uvw に分解できる
 1. $|uv| \leq n$
 2. $|v| \geq 1$
 3. 0 以上の任意の i について $uv^i w \in L$
- 条件を満たす分解が1つでもあればよい

ソフトウェアモデル論(2011/11/04)

4

反復補題は必要条件

(復習)

- 反復補題は正規言語の必要条件
 - 反復補題が成り立たなければ正規言語ではない
- 反復補題を満たす正規言語でない言語も存在する

ソフトウェアモデル論(2011/11/04)

5

レポートその5

- $L = \{ a^n \mid n = n^2, n \text{ は自然数} \}$ が正規言語でないことを反復補題を用いて証明
- L が正規言語であると仮定して、反復補題が成り立たないことを示す
- 反復補題が成り立たないとは...
 - 適当な正整数 n に対して $|z| \geq n$ なる z を選び、 z を $z = uvw$ (ただし $|uv| \leq n, |v| \geq 1$) となるように u, v, w に分解すると、どのように分解しても $uv^i w \notin L$ となる i が存在する

ソフトウェアモデル論(2011/11/04)

6

レポートその5

- L が正規言語であると仮定する
- 適当な正整数 n に対し a^n (ただし $m=n^2$) を選ぶ
- $a^m=uvw$ に分解
 - ただし $u=a^i, v=a^j, w=a^k$
 - また $i \geq 0, j \geq 1, k \geq 0, i+j \leq n, i+j+k=n^2$
- 反復補題より $uv^2w \in L$ のはずである
- 本当にそうか？

ソフトウェアモデル論(2011/11/04)

7

レポートその5

- $|uv^2w| = |a^i a^{2j} a^k| = i+2j+k$ である
- $j \geq 1$ および $i+j+k=n^2$ より $i+2j+k = n^2+j > n^2$
- $i \geq 0$ および $i+j \leq n$ より $j \leq n$
- よって $i+2j+k = n^2+j \leq n^2+n < n^2+2n+1 = (n+1)^2$
- つまり $n^2 < |uv^2w| < (n+1)^2$ なので $uv^2w \notin L$
- これは $uv^2w \in L$ に反する
- よって L は正規言語ではない

ソフトウェアモデル論(2011/11/04)

8

チューリング機械

ソフトウェアモデル論(2011/11/04)

9

チューリング機械

(復習)

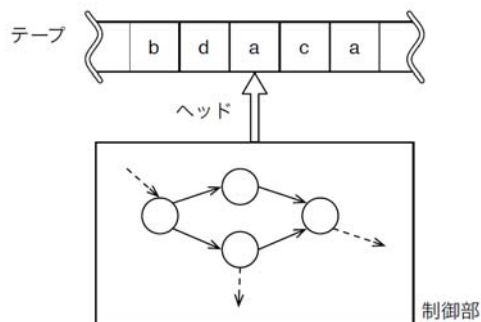
- Alan Turing, 1930's
- 計算を機械的動作としてとらえたモデル

ソフトウェアモデル論(2011/11/04)

10

概念図

(復習)



ソフトウェアモデル論(2011/11/04)

11

有限オートマトンとの違い

(復習)

- 制限なし
 - テープへの書き込みも可能
 - 1マス読んだらヘッドを
 - 右へ1マス移動
 - 左へ1マス移動
 - 移動しない

ソフトウェアモデル論(2011/11/04)

12

チューリング機械の動作 (復習)

1. ヘッドの位置のマス of 記号を読む
2. 読んだ記号に従って状態を遷移する
 - 終了状態へ到達したら終了
3. ヘッドの位置のマスに記号を書く
4. ヘッドの位置を
 - a. 1マス右へ移動する
 - b. 1マス左へ移動する
 - c. 移動しない
5. 1.へ戻る

ソフトウェアモデル論(2011/11/04) 13

チューリング機械の定義

$M = (Q, \delta, \Sigma, \Gamma)$

Q : 状態の有限集合 ($\neq \emptyset$)

$q_0 \in Q$: 初期状態

$q_{fin} \in Q$: 終了状態

δ : 遷移関数

$(Q - \{q_{fin}\} \times \Gamma) \rightarrow Q \times \Gamma \times \{L, R, N\}$

Σ : 入出力記号の有限集合

Γ : テープ記号の有限集合

ソフトウェアモデル論(2011/11/04) 14

テープ記号 Γ

- テープのマスに書くことができる記号のすべて
- 空白のマスを表す記号 B を含む

ソフトウェアモデル論(2011/11/04) 15

入出力記号 Σ

- チューリング機械の入出力に利用できる記号のすべて
- Γ の部分集合
- B は入出力に使えない
- 以下では $\Sigma = \{0, 1\}$ とする

ソフトウェアモデル論(2011/11/04) 16

状態遷移関数 δ

- $\delta : (Q - \{q_{fin}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$
- $\delta(q, a) = (q', b, L)$
 - 状態 q で記号 a を読んだ場合、
 1. 状態 q' に遷移し
 2. 記号 b を書き込み (a を上書きし)
 3. ヘッドを左へ1マス移動する
 - R ならば右へ1マス移動
 - N ならば移動しない
- 終了状態の遷移先はない

ソフトウェアモデル論(2011/11/04) 17

例: 関数 $inc(x) = x + 1$ を計算するTM

$M_{inc} = (\{0, fin, 1, 2\}, \delta, \{0, 1\}, \{0, 1, B\})$

状態	記号	遷移関数値
0	0	(0, 0, R)
0	1	(0, 1, R)
0	B	(1, B, L)
1	0	(2, 1, L)
1	1	(1, 0, L)
1	B	(fin, 1, N)
2	0	(2, 0, L)
2	1	(2, 1, L)
2	B	(fin, B, R)

$a = 0 \text{ or } 1$

ソフトウェアモデル論(2011/11/04) 18

チューリング機械の計算状況

- 計算中のチューリング機械の様子は、制御部の状態、テープの内容、ヘッドの位置で決まる

(q, ω, ω')

- q : 制御部の状態
- ω : ヘッドより左側のテープの内容
- ω' : ヘッドから右側のテープの内容 (ヘッド位置含む)

ソフトウェアモデル論(2011/11/04)

19

初期状況

$(q_0, \dots BB, xBB\dots)$

- $x \in \Sigma^*$ が入力
- 例えば、 $(q_0, \dots BB, 111000BB\dots)$
 - $(q_0, B, 111000B)$ あるいは
 - $(q_0, \epsilon, 111000)$ とも書く

ソフトウェアモデル論(2011/11/04)

20

終了状況

$(q_{fin}, \omega, \omega')$

- 終了状態に到達
- 正常終了状況
 - $(q_{fin}, \dots BB, yBB\dots)$
 - y が出力

ソフトウェアモデル論(2011/11/04)

21

チューリング機械の計算動作

- 計算状況を遷移関数に従って変えること

- $q \in Q, u, v \in \Sigma^*, a, b \in \Sigma$ とすると

$$(q, ub, av) \vdash \begin{cases} (q', u, ba'v) & \text{if } \delta(q, a) = (q', a', L) \\ (q', uba', v) & \text{if } \delta(q, a) = (q', a', R) \\ (q', ub, a'v) & \text{if } \delta(q, a) = (q', a', N) \end{cases}$$

- \vdash が1回の計算動作を表す

ソフトウェアモデル論(2011/11/04)

22

例: M_{inc} の計算動作

- 入力が101の場合

$(0, B, 101B) \vdash (0, B1, 01B) \vdash (0, B10, 1B)$

$\vdash (0, B101, B) \vdash (1, B10, 1B) \vdash (1, B1, 00B)$

$\vdash (2, B, 110B) \vdash (2, B, B110B) \vdash (fin, B, 110B)$

ソフトウェアモデル論(2011/11/04)

23

チューリング機械の計算

- 入力 x に対するチューリング機械の計算
 - x に対する初期状況から計算動作を繰り返す過程の総称

- 計算列
 - 計算動作に伴って変化する計算状況の列

- 計算の正常終了
 - 正常終了状況に到達

ソフトウェアモデル論(2011/11/04)

24

関数を計算するチューリング機械

- 関数 f を計算するチューリング機械 M
 - f は Σ^* 上の1変数(1引数)関数
- $x \in \text{dom}(f)$ ならば M に x を入力して実行すると $f(x)$ を出力して正常終了
- $x \notin \text{dom}(f)$ ならば M に x を入力して実行すると正常終了しない
- 結果を $M(x)$ と書く(出力以外に終了状況を含む)

ソフトウェアモデル論(2011/11/04)

25

例: $\text{inc}(x)$ 再び

- 正確には

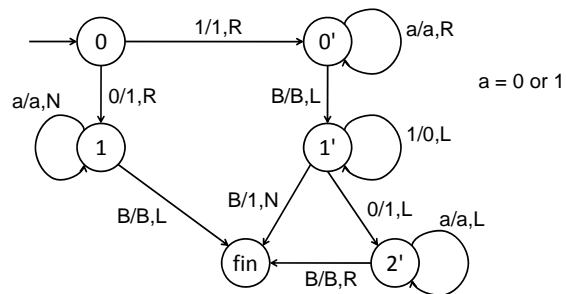
$$\text{inc}(x) = \begin{cases} n+1 \text{ の 2 進数表記} & \text{if } x \text{ がある } n \in \mathbb{N} \text{ の 2 進数表記} \\ \text{未定義} & \text{otherwise} \end{cases}$$
- x が正しい2進数でなければ未定義
 - 正しくない(定義域に含まれない)入力の場合は正常終了しない

ソフトウェアモデル論(2011/11/04)

26

例: $\text{inc}(x)$ 再び

$M^{\text{inc}} = (\{0, 1, \text{fin}, 0', 1', 2'\}, \delta, \{0, 1\}, \{0, 1, B\})$



ソフトウェアモデル論(2011/11/04)

27