

ソフトウェアモデル論(2010年度) 第6回・2010/11/01

桑原 寛明
情報理工学部 情報システム学科

レポートその3

- 別解もありえるので疑問があれば訊いて下さい
- NFAをDFAに変換
 - DFAの状態集合はNFAの状態集合のべき集合
 - $Q_N = \{S_0, S_1, S_2\}$ とすると
 - $Q_D = \{\emptyset, \{S_0\}, \{S_1\}, \{S_2\}, \{S_0, S_1\}, \{S_0, S_2\}, \{S_1, S_2\}, \{S_0, S_1, S_2\}\}$
 - 遷移図に状態 \emptyset を忘れないように注意
- $(a+b)^* \neq (a^*+b^*)$

ソフトウェアモデル論(2010/11/01)

2

正規言語

- 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

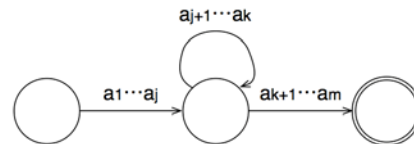
ソフトウェアモデル論(2010/11/01)

3

有限オートマトンが長い語を受理する場合

- 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(2010/11/01)

4

反復補題

- 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(2010/11/01)

5

$L = \{ a^n b^n \mid n \text{ は } 0 \text{ 以上の整数} \}$ は非正規言語

- L が正規言語と仮定する
- 適当な n に対して $|a^n b^n| \geq n$ となる l を選ぶ
- L は正規言語なので $a^l b^l = uvw$ と分解できる
- さらに、 $uv^2 w \in L$
- $a^l b^l$ の分解方法は以下の3通り
 1. $uvw = a^i a^j a^{k_1} b^{k_2}$ ($j \geq 1, k_1 \geq 0, i + j + k_1 = k_2 = l$)
 2. $uvw = a^i a^{j_1} b^{j_2} b^k$ ($j_1 \geq 1, j_2 \geq 1, i + j_1 = j_2 + k = l$)
 3. $uvw = a^i b^{j_1} b^{j_2} b^k$ ($j \geq 1, i_2 \geq 0, i_1 = i_2 + j + k = l$)

ソフトウェアモデル論(2010/11/01)

6

$L = \{ a^n b^n \mid n \text{は} 0 \text{以上の整数} \}$ は非正規言語

- 1. の場合、 $uv^2w = a^i a^{2j} a^{k_1} b^{k_2}$ であり、 $j \geq 1$ と $i + j + k_1 = k_2$ より $i + 2j + k_1 \neq k_2$ なので $uv^2w \notin L$
- 3. の場合も同様
- 2. の場合、 $uv^2w = a^i a^{j_1} b^{j_2} a^{j_1} b^{j_2} b^k$ であり、明らかに L の語ではない
- 以上より、 L は正規言語ではない
 - 反復補題の $uv^i w \in L$ を満たさない

ソフトウェアモデル論(2010/11/01)

7

チューリング機械

ソフトウェアモデル論(2010/11/01)

8

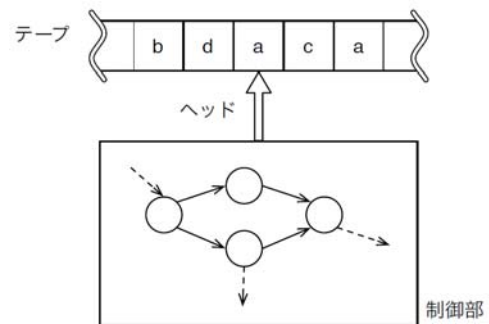
チューリング機械

- 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(2010/11/01)

9

概念図



ソフトウェアモデル論(2010/11/01)

10

有限オートマトンとの違い

- 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(2010/11/01)

11

チューリング機械の動作

1. ヘッドの位置のマスに記号を読む
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(2010/11/01)

12

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(2010/11/01)

13

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(2010/11/01)

14

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(2010/11/01)

15

状態遷移関数 δ

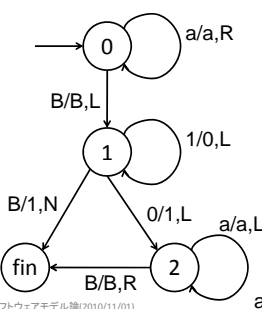
- $\delta : (Q - \{q_{fin}\}) \times \Gamma \rightarrow Q \times \Gamma \times \{L, R, N\}$
- $\delta(q, a) = (q', b, L)$
 - 状態 q で記号 a を読んだ場合、
 1. 状態 q' に遷移し
 2. 記号 b を書込み (a を上書きし)
 3. ヘッドを左へ1マス移動する
 - R ならば右へ1マス移動
 - N ならば移動しない
- 終了状態の遷移先はない

ソフトウェアモデル論(2010/11/01)

16

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(2010/11/01)

17

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(2010/11/01)

18

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(2010/11/01)

19

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(2010/11/01)

20

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(2010/11/01)

21

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(2010/11/01)

22

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(2010/11/01)

23

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(2010/11/01)

24

例: inc(x) 再び

- 正確には

$$inc(x) = \begin{cases} n+1 \text{ の 2 進数表記} & \text{if } x \text{ がある } n \in \mathbb{N} \text{ の 2 進数表記} \\ \text{未定義} & \text{otherwise} \end{cases}$$

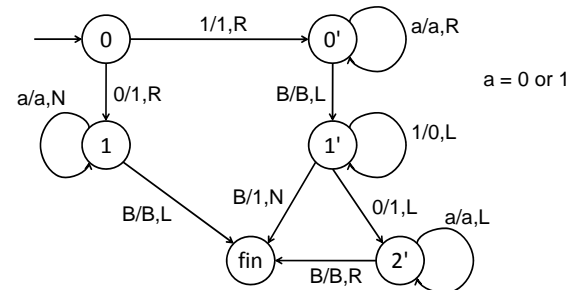
- x が正しい2進数でなければ未定義
 - 正しくない(定義域に含まれない)入力の場合は正常終了しない

ソフトウェアモデル論(2010/11/01)

25

例: inc(x) 再び

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



ソフトウェアモデル論(2010/11/01)

26

述語

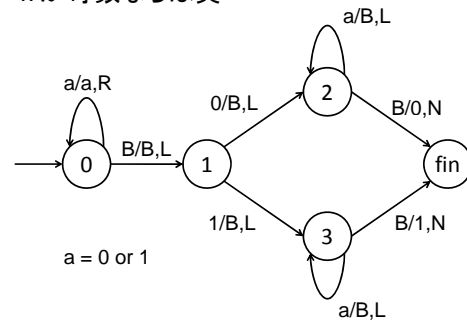
- すべての $x \in \Sigma^*$ に対して 0 または 1 を返す関数
 - 1 ならば真
 - 0 ならば偽
- 述語の定義域は Σ^* 全体なので、対応するTMは任意の入力に対して正常終了しなければならない
 - 終了状況が $(q_{fin}, B, 1B)$ または $(q_{fin}, B, 0B)$

ソフトウェアモデル論(2010/11/01)

27

例: odd(x)

- x が奇数ならば真



ソフトウェアモデル論(2010/11/01)

28

修正: 図2.7

- $\delta(1, B) = (fin, 0, N)$

ソフトウェアモデル論(2010/11/01)

29

多変数(多引数)関数の計算

- 引数の区切りを表す特殊な記号(資料では#)を利用して1変数関数として計算する
- 初期状況 $(q_0, B, x_1 \# x_2 B)$

ソフトウェアモデル論(2010/11/01)

30