

ソフトウェアモデル論(2010年度) 第7回・2010/11/08

桑原 寛明
情報理工学部 情報システム学科

反復補題について補足

- 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(2010/11/08)

2

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(2010/11/08)

3

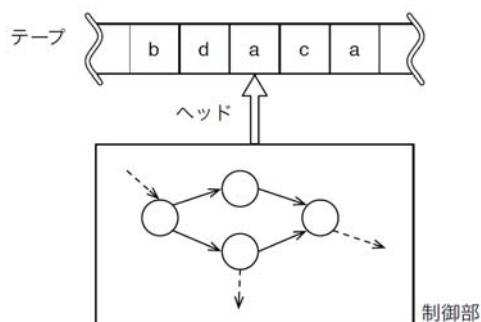
チューリング機械の動作

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(2010/11/08)

4

概念図



ソフトウェアモデル論(2010/11/08)

5

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(2010/11/08)

6

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(2010/11/08)

7

チューリング機械の計算

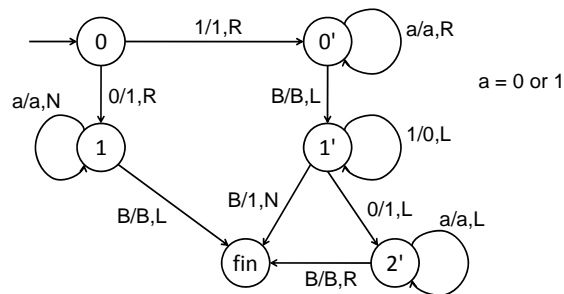
- 入力 x に対するチューリング機械の計算
 - x に対する初期状況から計算動作を繰り返す過程の総称
- 計算列
 - 計算動作に伴って変化する計算状況の列
- 計算の正常終了
 - 正常終了状況に到達

ソフトウェアモデル論(2010/11/08)

8

例: $\text{inc}(x) = x + 1$ を計算する TM M'_{inc}

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



ソフトウェアモデル論(2010/11/08)

9

例: M'_{inc} の計算動作

- 入力が101の場合
 - $(0, B, 101B) \vdash (0, B1, 01B) \vdash (0, B10, 1B)$
 - $\vdash (0, B101, B) \vdash (1, B10, 1B) \vdash (1, B1, 00B)$
 - $\vdash (2, B, 110B) \vdash (2, B, B110B) \vdash (\text{fin}, B, 110B)$

ソフトウェアモデル論(2010/11/08)

10

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(2010/11/08)

11

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 $(q_0, B, x_1\#x_2B)$

ソフトウェアモデル論(2010/11/08)

12

データのコード化

- 入出力記号の集合 Σ を $\{0, 1\}$ に制限する
- 制限しても困ることはない
 - すべてを2進数で表せばよい

ソフトウェアモデル論(2010/11/08)

13

自然数のコード化

- 例えば2進数
 - 0を除いて先頭が0でない2進数で表す
- $0_{10} = 0_2$
- $5_{10} = 101_2$
- $15_{10} = 1111_2$

ソフトウェアモデル論(2010/11/08)

14

英文字、数字、記号のコード化

- 例えばASCII
 - 8ビットで1記号
- $A \rightarrow 01000001$
- $1 \rightarrow 00110001$
- $(\rightarrow 00101000$

ソフトウェアモデル論(2010/11/08)

15

データの組や列のコード化

- $(2, AC)$ や $(1, 0, 0, 123)$ など
- 英文字、数字、カンマや括弧などの記号をすべてASCII化して並べる
- $(2, AC) \rightarrow 00101000\ 00110010\ 00101100\ \dots$
- $(1, 0, 0, 123) \rightarrow 00101000\ 00110001\ 00101100\ \dots$

ソフトウェアモデル論(2010/11/08)

16

2引数関数の1引数化

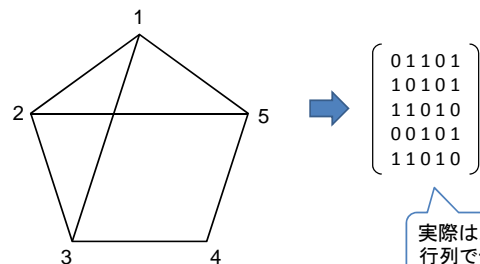
- 多引数を組にして1引数化する
- $\text{concat}(u, v)$
 - 組 (u, v) を引数にとる関数を考える
- $\text{concat}'(x)$
 - $x = (u, v)$ ならば $\text{concat}(u, v)$ を計算
 - それ以外ならば未定義

ソフトウェアモデル論(2010/11/08)

17

グラフのコード化

- 隣接行列を利用



ソフトウェアモデル論(2010/11/08)

18

チューリング機械のコード化

- チューリング機械 $M = (Q, \delta, \Sigma, \Gamma)$
 - Q, Σ, Γ は有限集合
 - δ は有限集合から有限集合への関数
- 有限 \Rightarrow 列挙可能 \Rightarrow コード化可能
- $(Q, \delta, \Sigma, \Gamma)$ をどのようにコード化するか

ソフトウェアモデル論(2010/11/08)

19

Q, Σ, Γ のコード化

- 状態は $0, 1, 2, \dots, n$ と名前が付いており n を終了状態とすれば Q は状態数で決定できる
- Σ は $\{0, 1\}$
- 記号を $0, 1, 2, \dots, m$ (ただし 2 を B とみなす) とすれば Γ は記号数で決定できる

ソフトウェアモデル論(2010/11/08)

20

δ のコード化

- 遷移関数は有限集合から有限集合への関数
 - 状態 \times 記号 \rightarrow 状態 \times 記号 \times 移動方向
 - 有限の対応付け
- δ の引数の状態番号の小さい順、記号番号の小さい順に δ の値を並べる

ソフトウェアモデル論(2010/11/08)

21

δ のコード化の例

- M_{inc} の遷移関数

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)

順に並べる

(0,0,R), (0,1,R), (1,B,L),
(2,1,L), (1,0,L), ...

ソフトウェアモデル論(2010/11/08)

22

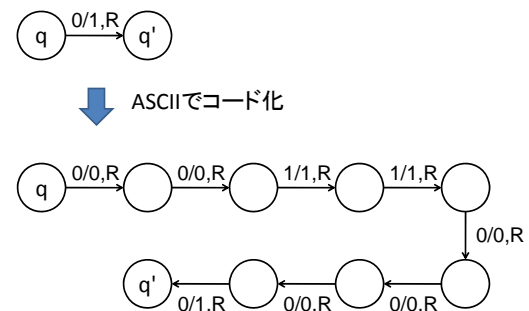
例: M_{inc} のコード化

- 状態数 4
 - $\{0, 1, 2, fin\}$
- 記号数 3
 - $\{0, 1, B\}$
- 遷移関数
 - $(0,0,R), (0,1,R), (1,B,L), (2,1,L), \dots, (fin,B,R)$
- よって
 - $(4, 3, (0,0,R), (0,1,R), (1,2,L), \dots, (3,2,R))$

ソフトウェアモデル論(2010/11/08)

23

コード化に伴うTMの変形例(その1)



ソフトウェアモデル論(2010/11/08)

24

コード化に伴うTMの変形例(その2)

ソフトウェアモデル論(2010/11/08) 25

テープ記号は 0, 1, B で十分

- すべてを2進数でコード化すればよい
- 0, 1 で表現できる
- テープは無限長なので B は必要
- 多引数関数の引数は列で表して1引数とする

ソフトウェアモデル論(2010/11/08) 26

合成関数の計算

- 合成関数 $g \circ f$ を計算するTM
- 関数 f を計算するTM M_f と g を計算するTM M_g から生成
- 直観的には、 M_f の終了状態を M_g の初期状態とみなせばよい

ソフトウェアモデル論(2010/11/08) 27

修正: 定理2.3

- 61ページ、2行目
- $f \circ g \rightarrow g \circ f$
- $\delta_3(q,a) = (q_0', a, N)$ ← 内側の()は不要

ソフトウェアモデル論(2010/11/08) 28

分岐

- 述語 $p?$ と関数 f_1, f_2 に対して以下を計算

$$f(x) = \begin{cases} f_1(x) & \text{if } p?(x) = 1 \\ f_2(x) & \text{if } p?(x) = 0 \end{cases}$$

- $p?$ を計算した後、その結果を見て f_1 を計算するTM と f_2 を計算するTM のいずれの初期状態に遷移するか決める

ソフトウェアモデル論(2010/11/08) 29

分岐

ソフトウェアモデル論(2010/11/08) 30

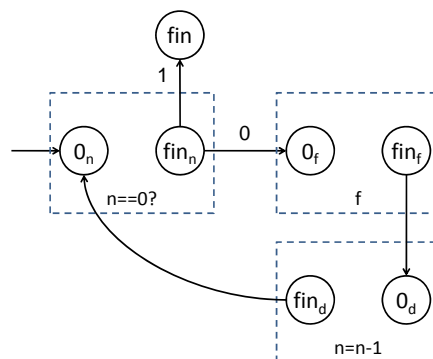
繰返し

- 関数 f に対して $g(x,n) = f^{(n)}(x)$ を計算する
 - $f^{(n)}$ は f を n 回合成した関数
- $n=0$ による条件分岐と f の計算、 $n=n-1$ の計算を組み合わせればよい

ソフトウェアモデル論(2010/11/08)

31

繰返し



ソフトウェアモデル論(2010/11/08)

32