ソフトウェアモデル論(2010年度)
第5回・2010/10/25

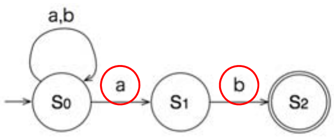
桑原 寛明
情報理工学部 情報システム学科

レポートの略解

- 締切を過ぎたものから順次 WebCT に掲載

有限オートマトンが受理する言語の特徴？

- 受理する言語
  - 初期状態から最終状態に到達する語の集合
- 受理する言語の特徴
  - 受理される語すべてに共通な特徴



最終状態には必ず a, b の順で遷移して到達

ab で終了する語は必ず受理される

インタラクティブシート

- 講義の終わりで
- 独自設問(Q7)：進度について
  1. もっと速くてもよい
  2. 今ぐらいがよい
  3. もう少しゆっくりがよい
- 講義前にスライドの公開はしません
- 演習問題を後日配布する予定

有限オートマトンから正規表現への変換

- 有限オートマトン $M = (\{1,\ldots,n\}, \Sigma, \delta, 1, F)$
- r_{ij}^{k} を求める
  - 状態 i から状態 j へ k 以下の状態のみを通過して到達する記号列の正規表現
  - r_{ij}^{0} から順に帰納的に
- $r_{1f1}^{n} + \ldots + r_{1fl}^{n}$ が初期状態から受理状態へ到達する記号列の正規表現
  - $\{ f1, \ldots, fl \}$ は受理状態の集合

---

r_{ij}^{0} を求める

- 状態 i から状態 j へ直接到達
- $\delta(i,a) = j$ を満たす a の集合を $\{ a_1, \ldots, a_l \}$ とする
  - そのような a がなければ $\emptyset$
- $i \neq j$ ならば $r_{ij}^{0} = a_1 + \ldots + a_l$
- $i = j$ ならば $r_{ij}^{0} = a_1 + \ldots + a_l + \varepsilon$

r_{ij}^k を求める

- $r_{ij}^k = r_{ik}^{k-1} (r_{kk}^{k-1})^* r_{kj}^{k-1} + r_{ij}^{k-1}$
- 状態 k を通過するかしないかの2択
  - 通過する
    1. i から初めて k に到達
    2. k-1 以下のみ通過してまた k に到達（を繰り返す）
    3. k から k-1 以下を通過して j に到達
  - 通過しない
    - k-1 以下の状態のみ通過

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

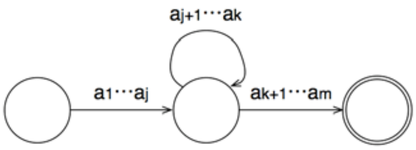
ある言語が正規言語か否か判定するにはどうすればよいか

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

部屋割り原理
鳩ノ巣原理

- 状態数が n で語の長さが n 以上だと…



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 $uv^iw \in L$
- 条件を満たす分解が1つでもあればよい

L={ a^nb^n | nは0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 $|a^lb^l| \geq n$ となる l を選ぶ
- L は正規言語なので a^lb^l = uvw と分解できる
- さらに、$uv^2w \in L$
- a^lb^l の分解方法は以下の3通り
  1. $uvw = a^ia^ja^{k1}b^{k2}$ $(j \geq 1, k_1 \geq 0, i + j + k_1 = k_2 = l)$
  2. $uvw = a^ia^{j1}b^{j2}b^k$ $(j_1 \geq 1, j_2 \geq 1, i + j_1 = j_2 + k = l)$
  3. $uvw = a^{i1}b^{i2}b^jb^k$ $(j \geq 1, i_2 \geq 0, i_1 = i_2 + j + k = l)$

L={ a^nb^n | nは0以上の整数 } は非正規言語

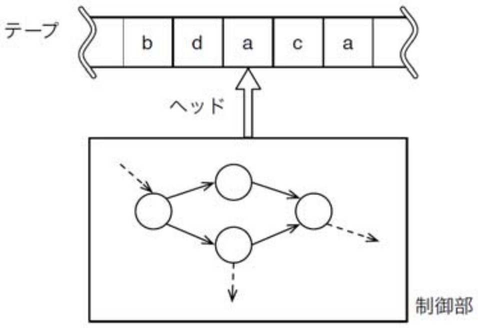
- 1. の場合、$uv^2w = a^ia^{2j}a^{k1}b^{k2}$ であり、$j \geq 1$ と $i + j + k_1 = k_2$ より $i + 2j + k_1 \neq k_2$ なので $uv^2w \notin L$
- 3. の場合も同様
- 2. の場合、$uv^2w = a^ia^{j1}b^{j2}a^{j1}b^{j2}b^k$ であり、明らかに L の語ではない
- 以上より、L は正規言語ではない
  - 反復補題の $uv^iw \in L$ を満たさない

チューリング機械

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

概念図



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

チューリング機械の動作

1. ヘッドの位置のマスの記号を読む
2. 読んだ記号に従って状態を遷移する
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

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

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

Σ : 入出力記号の有限集合

Γ : テープ記号の有限集合

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

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 Σ = { 0, 1 } とする

状態遷移関数 δ

- δ(q, a) = (q', b, L)
  - 状態 q で記号 a を読んだ場合、
    状態 q' に遷移し
    記号 b を書込み（a を上書きし）
    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