第 11 講 FSA が定義する言語

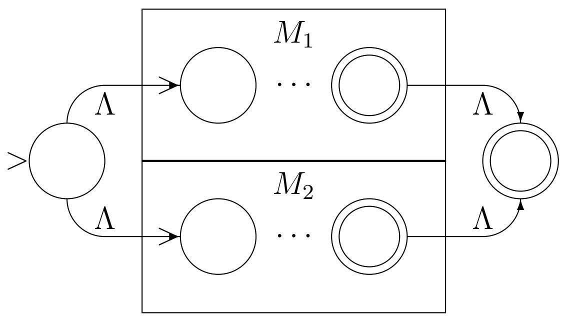
林　恒俊

FSA が定義する言語の性質

- ここまでの議論で FSA が定義する言語は正規言語であることが推測される。以下でこれが正しいかどうかについて検討する。
- まず最初に FSA が定義する言語の性質について調べる。そして性質が明確になれば正規言語との関係についても容易に理解することができる。
- 当然 FSA は DFA と NFA 双方全体を含むものとして考える。

言語の和

- 2 個の FSA M_1, M_2 について次のような FSA $M_\cup$ を考える。



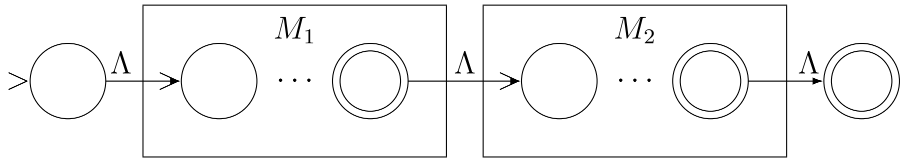
- $M_\cup$ は M_1, M_2 に
  - 新たに初期状態と終了状態を追加して
  - 追加した初期状態から空遷移で M_1, M_2 の初期状態に遷移し
  - M_1, M_2 のすべての終了状態から追加した終了状態へ空遷移で遷移する

ように構成される。なお $M_\cup$ は NFA である。

- $M_\cup$ の受理する言語 $L(M_\cup) = L(M_1) \cup L(M_2)$ である。なぜか。
- 以上のように $L_1 \in \mathcal{L}_{\mathbf{FSA}}$ かつ $L_2 \in \mathcal{L}_{\mathbf{FSA}}$ なら $L_1 \cup L_2 \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は和演算について閉じているという。
- FSA が 2 個以上の場合でも上記の手段を繰返して適用すればそれらが定義する言語の和を受理する FSA を構築することができる。

言語の連結

- M_1, M_2 を 2 個の FSA とする。次のような FSA $M_\circ$ を考える。

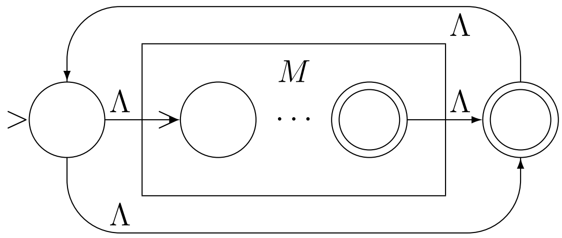


- $M_\circ$ は M_1, M_2 に
  - 新たに初期状態と終了状態を追加して
  - 追加した初期状態から M_1 の初期状態に空遷移で遷移し
  - M_1 のすべての終了状態から M_2 の初期状態に空遷移で遷移し
  - M_2 のすべての終了状態から追加した終了状態へ空遷移で遷移する

  ように構成される。なお形式的に $M_\circ$ は NFA である。
- $M_\circ$ の受理する言語 $L(M_\circ) = L(M_1) \circ L(M_2)$ である。なぜか。
- 以上のように $L_1 \in \mathcal{L}_{\mathbf{FSA}}$ かつ $L_2 \in \mathcal{L}_{\mathbf{FSA}}$ なら $L_1 \circ L_2 \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は連結演算について閉じているという。
- FSA が 2 個以上の場合でも上記の手段を繰返して適用すればそれらが定義する言語の連結を受理する FSA を構築することができる。

Kleene の $*$ 演算

- M を FSA とする。次のような FSA M_* を考える。



- M_* は M に
  - 新たに初期状態と終了状態を追加し
  - 追加した初期状態から M の初期状態に空遷移で遷移し
  - M のすべての終了状態から追加した終了状態へ空遷移で遷移し
  - さらに追加した初期状態と終了状態が互いに空遷移で遷移する

  ように構成される。なお M_* は NFA である。
- M_* の受理する言語 $L(M_*) = L(M)^*$ である。なぜか。
- 以上の検討から $L \in \mathcal{L}_{\mathbf{FSA}}$ なら $L^* \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は Kleene の $*$ 演算について閉じているという。

補集合

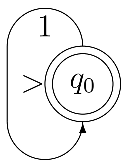
- 任意の NFA について同一言語を受理する DFA が構成可能である。したがって DFA について考察すればその結果は FSA 全体について適用できる。
- $M = (Q, \Sigma, \delta, q_0, F)$ で定義された DFA M が受理する言語を L とする。ただし $Q = \{q_0, q_1, \ldots, q_n\}$ である。次のような手順で DFA M_c を構成する。
- まず M を完全化して次のように M_T を構成する。
  - M に未定義遷移が存在する場合 M に新規に状態 q' を追加する

- ただし q' は終了状態集合には含めない ($q' \notin F$)
- M の未定義状態遷移 (q_i, σ) について (q_i, σ, q') を δ に追加する
- すべての $\sigma \in \Sigma$ について遷移 (q', σ, q') を δ に追加する
- 以上により完全化が完成し M_T が構成される

なお完全化しても受理する言語はかわらない。

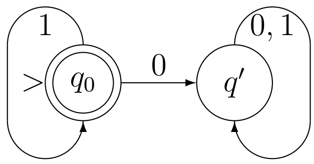
- 次に M_T の終了状態と非終了状態を交換した DFA を M_c とする。M_c は M が受理した記号列を拒否し、拒否した記号列を受理する。すなわち L^c を受理する。
- アルファベット $\Sigma = \{0, 1\}$ 上の言語 $\{1\}^* = \{\Lambda, 1, 11, \ldots\}$ を受理する DFA M_1 は次のように構成される。これを完全化して終了状態と非終了状態を入換えた DFA は確かに $\{1\}^*$ の補集合 $\{x \mid \#_0(x) \geq 1\}$ を受理する。

M_1



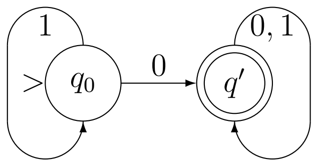
$L(M_1) = \{1\}^*$

M_T



$$L(M_T) = L(M_1) = \{1\}^*$$

M_c



$$L(M_c) = L(M_1)^c = \Sigma^* - \{1\}^*$$

- 以上の議論から $L \in \mathcal{L}_{\mathbf{FSA}}$ なら $L^c \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は補集合演算について閉じているという。

逆列言語

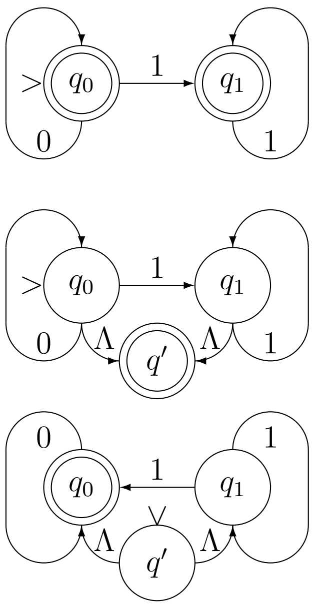
- 言語 L を受理する FSA M から次のような NFA M_R を構成する。

- M に新たに状態 q' を追加し M のすべての終了状態から q' に空遷移するようにした FSA M' を構成する。すなわちすべての $f \in F$ について (f, Λ, q') を遷移関数に加える。M' の終了状態は q' のみとする。
- この操作は M の終了状態を M' の 1 個の状態 q' に統合する。
- M' の遷移関数のすべての要素 (q_1, σ, q_2) の q_1 と q_2 を交換する。すなわち遷移の方向を逆向きにする。
- 初期状態と終了状態を入替えて M_R とする。M_R の初期状態は q' で終了状態は q_0 である。
- M_R は M が受理する記号列の逆列を受理する。なぜか。
- 以上の検討から $L \in \mathcal{L}_{\mathbf{FSA}}$ なら $L^R \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は逆列演算について閉じているという。
- アルファベット $\Sigma = \{0, 1\}$ 上の言語 $\{0\}^*\{1\}^*$ を受理する DFA M_{01} は次のように構成される。M の終了状態を統合し遷移を反転した NFA M_R は $\{0\}^*\{1\}^*$ の逆列言語 $\{1\}^*\{0\}^*$ を確かに受理する。

M_{01} $\qquad$ $L(M_{01}) = \{0\}^*\{1\}^*$

M' $\qquad$ $L(M') = L(M_{01}) = \{0\}^*\{1\}^*$

M_R $\qquad$ $L(M_R) = L(M')^R = \{1\}^*\{0\}^*$



積 (共通) 集合

- $L_1 \in \mathcal{L}_{\mathbf{FSA}}$ かつ $L_2 \in \mathcal{L}_{\mathbf{FSA}}$ とする。

- DeMorgan の定理により $L_1 \cap L_2 = (L_1^c \cup L_2^c)^c$ である。$\mathcal{L}_{\mathbf{FSA}}$ は和集合と補集合について閉じているため $L_1 \cap L_2 \in \mathcal{L}_{\mathbf{FSA}}$ である。これを $\mathcal{L}_{\mathbf{FSA}}$ は積集合演算について閉じているという。
- この議論は正しい。けれども積集合を受理する FSA がどのようなものか具体的に明確に与えてくれない。次のような**構成的** (**constructive**) 証明が望ましい場合もある。

積集合に関する性質の構成的証明

- ある入力記号列について複数の DFA を並列的に動作させる機械を構成することにより証明する。すべての DFA が受理する記号列は積集合の要素と見なされるからである。
- 2 個の DFA M_1 と M_2 から次のような DFA $M_\cap$ を合成する。ただし

$$
\begin{aligned}
M_1 &= (Q, \Sigma, \delta_1, q_0, F_1) \quad Q = \{q_0, q_1, \ldots, q_m\} \\
M_2 &= (R, \Sigma, \delta_2, r_0, F_2) \quad R = \{r_0, r_1, \ldots, r_n\}
\end{aligned}
$$

- $M_\cap$ の状態は M_1 と M_2 の状態の対 (q_i, r_k) とする。すなわち $M_\cap$ の状態集合 $Q' = Q \times R$ である。
- $M_\cap$ の初期状態は (q_0, r_0) とする。
- $M_\cap$ の終了状態集合は $F_1 \times F_2$ とする。
- M_1 の状態 $q_i, q_j \in Q_1$ と M_2 の状態 $r_k, r_l \in Q_2$ について

  $$(q_i, \sigma, q_j) \in \delta_1 \text{ かつ } (r_k, \sigma, r_l) \in \delta_2$$

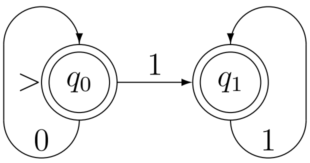
  なら $M_\cap$ の遷移関数 δ' に $((q_i, r_k), \sigma, (q_j, r_l))$ を追加する。この遷移は (q_i, σ, q_j) と (r_k, σ, r_l) を並行して遷移するのと同じである。
- このように構成された $M_\cap$ は任意の記号列 x について
  - ◦ $x \in L(M_1) \wedge x \in L(M_2)$ なら $x \in L(M_\cap)$
  - ◦ $x \in L(M_\cap)$ なら $x \in L(M_1) \wedge x \in L(M_2)$

  である。すなわち $L(M_\cap) = L(M_1) \cap L(M_2)$ である。なぜか。

積集合言語 DFA の例

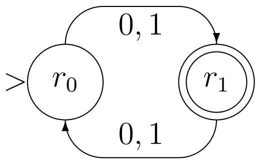
- アルファベット $\{0,1\}$ 上の次 DFA M_{01} と M_{odd} について積集合言語を受理する DFA $M_\cap$ を構築する。$L(M_{01}) = \{0\}^*\{1\}^*$ で $L(M_{\mathrm{odd}}) = \{x \mid x \in \{0,1\}^*, |x|$ は奇数 $\}$ である。

M_{01}



$$
\begin{aligned}
M_{01} &= (\{q_0, q_1\}, \{0,1\}, \delta_{01}, q_0, F_{01}) \\
\delta_{01} &= \{(q_0, 0, q_0), (q_0, 1, q_1), \\
&\quad (q_1, 1, q_1)\} \\
F_{01} &= \{q_0, q_1\}
\end{aligned}
$$

M_{odd}



$$
\begin{aligned}
M_{\mathrm{odd}} &= (\{r_0, r_1\}, \{0,1\}, \delta_{\mathrm{odd}}, r_0, F_{\mathrm{odd}}) \\
\delta_{\mathrm{odd}} &= \{(r_0, 0, r_1), (r_0, 1, r_1), \\
&\quad (r_1, 0, r_0), (r_1, 1, r_0)\} \\
F_{\mathrm{odd}} &= \{r_1\}
\end{aligned}
$$

- DFA $M_\cap = (Q', \{0,1\}, \delta', q_0', F')$ とする。

- 0 による M_{01} の遷移は $(q_0, 0, q_0)$ で M_{odd} の遷移は $(r_0, 0, r_1)$ と $(r_1, 0, r_0)$ である。したがって 0 による $M_\cap$ の遷移は

$$((q_0, r_0), 0, (q_0, r_1)), ((q_0, r_1), 0, (q_0, r_0))$$

  となる。

- 1 による M_{01} の遷移は $(q_0, 1, q_1)$ と $(q_1, 1, q_1)$ で M_{odd} の遷移は $(r_0, 1, r_1)$ と $(r_1, 1, r_0)$ である。したがって 1 による $M_\cap$ の遷移は

$$
\begin{gathered}
((q_0, r_0), 1, (q_1, r_1)), ((q_0, r_1), 1, (q_1, r_0)), \\
((q_1, r_0), 1, (q_1, r_1)), ((q_1, r_1), 1, (q_1, r_0))
\end{gathered}
$$

  となる。

- 遷移関数 δ' はこれらのすべての遷移を要素とする。

- 状態集合 $Q' = \{(q_0, r_0), (q_0, r_1), (q_1, r_0), (q_1, r_1)\}$

- 初期状態 $q_0' = (q_0, r_0)$

- 終了状態集合 $F' = \{(q_0, r_1), (q_1, r_1)\}$

- 以上から $M_\cap$ は次の状態遷移図で与えられる。