第 8 講 有限状態機械の設計

林　恒俊

DFA の実例と設計

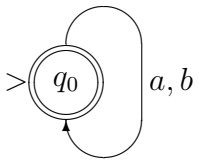
- DFA のような抽象機械は形式言語理論を構成する数学的道具である。しかし見方を変えればこれらの機械はハードウェア・ソフトウェア両面に渡る現実的実体のモデルと見なすことができる。
- DFA に関する考察は自動制御の制御装置設計等に応用することができる。自動洗濯乾燥機や自動車の自動変速装置を考えるとよい。また計算機の CPU は非常に複雑な DFA そのものである。
- DFA についてより高度な理解を得るためここではいくつかの言語についてそれを受理する DFA を設計する。
- DFA の定義からそれが受理する言語を推測することはそれほど容易ではない。原理的には言語を求めるためには DFA の状態遷移図で開始状態から終了状態に到達する枝を辿って受理する記号列を求めるとよい。状態遷移図が複雑なループを含む場合にはこれは容易ではない。
- 逆に言語から DFA を設計する場合も記号列に適合するように状態遷移図を構成するとよい。
- なお以下の例題から FSA が定義する言語は正規言語・正規表現と関連が深いことが推測される。

すべての記号列を受理する DFA

- アルファベット $\{a, b\}$ 上のすべての記号列 $\{a, b\}^*$ を受理する DFA M_* は形式的に次のように定義される。

$$M_* = (\{q_0\}, \{a, b\}, \{(q_0, a, q_0), (q_0, b, q_0)\}, q_0, \{q_0\})$$

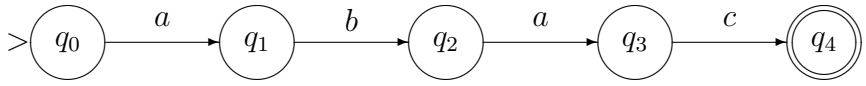
- M_* は次のような状態遷移図で表示される。



  実際に M_* が $\{a, b\}$ 上のすべての記号列を受理するかどうか考察せよ。

記号列パターンを検出する DFA

- アルファベット $\Sigma = \{a, b, c\}$ 上の記号列で $abac$ を部分列として含む記号列を検出する DFA M_{abac} を考える。なお部分列として $abac$ を含む記号列は正規表現 `(a|b|c)*abac(a|b|c)*`で与えられる。
- DFA の基本的な構造として次のような機械 M^0_{abac} を考える。M^0_{abac} は記号列の先頭から a, b, a, c の順に記号を走査して記号列が検出できれば終了する。



  M^0_{abac} は

$$M^0_{abac} = (\{q_0, q_1, q_2, q_3, q_4\}, \{a, b, c\}, \delta, q_0, \{q_4\})$$
$$\delta = \{(q_0, a, q_1), (q_1, b, q_2), (q_2, a, q_3), (q_3, c, q_4)\}$$

  と定義され

$$L(M^0_{abac}) = \{abac\}$$

  である。

- このままでは $abac$ 以外の記号列を認識しないため M^0_{abac} に次のような修正を加えて M_{abac} を構成する。
  - 走査中にに $abac$ でない記号列があらわれた時再試行を行うように状態遷移を追加する。
  - 一旦 $abac$ を検出した後はどのような記号列があらわれてもよいため残りの記号列を読み飛ばす状態遷移を追加する。

- 検出後に追加する状態遷移は自明である。どの記号があらわれても q_4 に戻ればよいので

$$\{(q_4, a, q_4), (q_4, b, q_4), (q_4, c, q_4)\}$$

を δ に追加する。

- 前半に追加する状態遷移については記号列と状態を慎重に検討する必要がある。

  - q_0 で $\{b, c\}$ が現れた場合

    まだ探索記号列 $abac$ を検出していないことになるので q_0 に戻ればよい。状態遷移

$$\{(q_0, b, q_0), (q_0, c, q_0)\}$$

    を追加する。

  - q_1 で a があらわれた場合

    すでに q_1 では a を検出しているため q_1 に戻って走査を再開するとよい。

$$\{(q_1, a, q_1)\}$$

    を追加する。

  - q_1 で c があらわれた場合

    ここまでに検出した記号列はすべて無効になるので最初から走査を再開するために q_0 に遷移する

$$\{(q_1, c, q_0)\}$$

    を追加する。

  - q_2 で b あるいは c があらわれた場合

    検出した記号列は abb と abc であり $abac$ とは適合しない。最初から走査を再開することになるので

$$\{(q_2, b, q_0), (q_2, c, q_0)\}$$

    を追加する。

- q_3 で a があらわれた場合

  検出した記号列は $abaa$ となる。最後尾の a が $abac$ の先頭になるかもしれないので a を検出後の状態 q_1 へ遷移する

  $$\{(q_3, a, q_1)\}$$

  を追加する。

- q_3 で b があらわれた場合

  検出した記号列は $abab$ であり、その接尾辞 ab が $abac$ の接頭辞になる可能性があるので ab 検出後の状態 q_2 へ遷移する

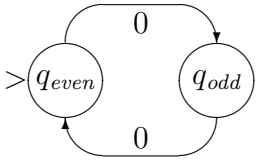
  $$\{(q_3, b, q_2)\}$$

  を追加する。

- 以上の状態遷移を追加した DFA はかなり複雑である。状態遷移図で表現してみるとよい。
- $abac$ を接頭辞とする $\{a, b, c\}$ 上の記号列を受理する DFA を設計してみよ。あるいは $abac$ を接尾辞とする記号列を受理する DFA も考察せよ。

列記号の偶数奇数を判別する DFA

- アルファベット $\Sigma = \{0, 1\}$ 上の入力記号列について 0 と 1 の個数がそれぞれ偶数または奇数の記号列を判別する DFA を考察する。DFA は記号列を先頭から走査する。
- 状態 q_{even}, q_{odd} の間で記号 0 について $\{(q_{even}, 0, q_{odd}), (q_{odd}, 0, q_{even})\}$ という遷移を行うものとする。



初期状態が q_{even} ならこの DFA は入力記号列中の 0 の数を数えて

- 偶数個の時状態 q_{even} をとり
- 奇数個の時状態 q_{odd} をとる

(何故か)

- 同様に 1 についても偶数個のときに取る状態と奇数個のときに取る状態の 2 状態が考えられる。
- 2 記号についてそれぞれが偶数個と奇数個のどちらかしかありえないので取りうる状態は

	0 偶数個	0 奇数個
1 偶数個	q_0	q_1
1 奇数個	q_3	q_2

の組合わせの 4 個である。4 状態あれば 2 個の記号の個数について偶数奇数の判別が可能である。

- 記号列の長さが 0 の時偶数と見なされるので初期状態を q_0 に取ればよい。
- 状態 q_0 と q_1 の間では入力が 0 で互いに遷移する。遷移関数に

$$\{(q_0, 0, q_1), (q_1, 0, q_0)\}$$

を追加する。同様に q_1 と q_2 間では 1、q_2 と q_3 間では 0、q_3 と q_0 間では 1 で互いに遷移するので

$$\{(q_1, 1, q_2), (q_2, 1, q_1), (q_2, 0, q_3), (q_3, 0, q_2), (q_3, 1, q_0), (q_0, 1, q_3)\}$$

を関数に追加すればよい。

- q_0 と q_2 間、q_1 と q_3 間では 1 個の記号で状態遷移することはない。1 個の記号入力で 2 個の記号数が同時に偶数 $\leftrightarrow$ 奇数に変わることはないからである。
- 終了状態を q_0, q_1, q_2, q_3 の中から適切に選択することにより入力記号中の記号の個数の偶数奇数を判別する DFA を構成できる。つぎの DFA M_{even} は入力記号がともに偶数個のものを受理する。

$$\begin{aligned} M_{even} &= (\{q_0, q_1, q_2, q_3\}, \{0, 1\}, \delta, q_0, \{q_0\}) \\ \delta &= \{(q_0, 0, q_1), (q_1, 0, q_0), (q_1, 1, q_2), (q_2, 1, q_1), \\ &\quad (q_2, 0, q_3), (q_3, 0, q_2), (q_3, 1, q_0), (q_0, 1, q_3)\} \end{aligned}$$