

「オートマトンと形式言語」補足資料(6)

1 文脈自由文法 (CFG) の例

例 1 CFGとして,

$$\mathcal{G}: S \rightarrow 1S1 \mid \#$$

を考える. 導出の例をいくつか示す:

$$\begin{aligned} S &\Rightarrow \# \\ S &\Rightarrow 1S1 \Rightarrow 1\#1 \\ S &\Rightarrow 1S1 \Rightarrow 11S11 \Rightarrow 11\#11 \end{aligned}$$

このとき, $L(\mathcal{G}) = \{1^n\#1^n \mid n \geq 0\}$ となる.

例 2 CFGとして,

$$\mathcal{G}: S \rightarrow 0S11 \mid 00$$

を考える. 導出の例をいくつか示す:

$$\begin{aligned} S &\Rightarrow 00 \\ S &\Rightarrow 0S11 \Rightarrow 00011 \\ S &\Rightarrow 0S11 \Rightarrow 00S1111DFA \Rightarrow 00001111 \end{aligned}$$

このとき, $L(\mathcal{G}) = \{0^{n+2}1^{2n} \mid n \geq 0\}$ となる.

例 3 CFGとして,

$$\begin{aligned} \mathcal{G}: S &\rightarrow 0S1 \mid T \\ T &\rightarrow 0T \mid \varepsilon \end{aligned}$$

を考える. 導出の例をいくつか示す:

$$\begin{aligned} S &\Rightarrow T \Rightarrow \varepsilon \\ S &\Rightarrow T \Rightarrow 0T \Rightarrow 0\varepsilon = 0 \\ S &\Rightarrow T \Rightarrow 0T \Rightarrow 00T \Rightarrow 00 \\ S &\Rightarrow 0S1 \Rightarrow 0T1 \Rightarrow 01 \\ S &\Rightarrow 0S1 \Rightarrow 0T1 \Rightarrow 00T1 \Rightarrow 001 \\ S &\Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 00T11 \Rightarrow 0011 \\ S &\Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 00T11 \Rightarrow 000T11 \Rightarrow 00011 \end{aligned}$$

このとき, $L(\mathcal{G}) = \{0^i1^j \mid i \geq j\}$ となる.

例 4 CFGとして,

$$\begin{aligned} \mathcal{G}: \quad S &\rightarrow S\#S \mid T \\ T &\rightarrow 0T1 \mid 01 \end{aligned}$$

を考える. 導出の例をいくつか示す:

$$\begin{aligned} S &\Rightarrow T \xrightarrow{*} 0^31^3 \\ S &\Rightarrow S\#S \Rightarrow T\#S \xrightarrow{*} 0^31^3\#S \Rightarrow 0^31^3\#T \xrightarrow{*} 0^31^3\#0^21^2 \\ S &\Rightarrow S\#S \Rightarrow S\#S\#S \xrightarrow{*} T\#T\#T \xrightarrow{*} 01\#0^21^2\#0^31^3 \end{aligned}$$

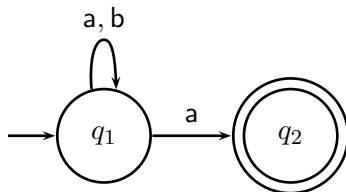
このとき, $L(\mathcal{G}) = \{0^{i_1}1^{i_1}\#0^{i_2}1^{i_2}\#\dots\#0^{i_k}1^{i_k} \mid k \geq 1, i_1, \dots, i_k \geq 1\}$ となる.

2 NFA から文脈自由文法への変換

NFA から等価な文脈自由文法は, 教科書 p.122にある, DFA から作るやり方と同じ. つまり, 状態遷移図から,

1. 状態を変数にとり, 開始状態を開始変数にする.
2. それぞれの状態遷移 $q_i \xrightarrow{a} q_j$ について, $q_i \rightarrow aq_j$ を生成規則に追加する.
3. それぞれの受理状態 $q_i \in F$ について, $q_i \rightarrow \varepsilon$ を生成規則に追加する.

例 5 アルファベットを $\Sigma = \{a, b\}$ とする. 以下の状態遷移図で表わされる NFA N_1 を考えてみる.



このとき, q_1, q_2 を変数にとり, 以下のような CFG を構成すればよい.

$$\begin{aligned} q_1 &\rightarrow aq_1 \mid bq_1 \mid aq_2 \\ q_2 &\rightarrow \varepsilon \end{aligned}$$

DFA や NFA が与えられれば, 等価な CFG が構成できるので, 正規言語は, 文脈自由言語でもある. 一方, 例 1 の文脈自由言語のように, 文脈自由言語のなかには正規言語でないものもある. したがって, 以下のような関係が成り立つ.

定理 6 正規言語のクラスは, 文脈自由言語のクラスの真部分クラスである.

3 文脈自由言語のクラスの閉包性

文脈自由言語のクラスは, 正規言語のクラスと同様に, 和集合, 連結演算, スター演算について閉じている. 一方, 正規言語のクラスとは対照的に, 積集合と補集合については, 閉じていない.

和集合に関する閉包性 2つの CFG \mathcal{G}_1 と \mathcal{G}_2 が与えられたとする.

$$\mathcal{G}_1: \quad S_1 \rightarrow \dots$$

$$\mathcal{G}_2: \quad S_2 \rightarrow \dots$$

このとき, 変数は適当に変更して, \mathcal{G}_1 と \mathcal{G}_2 では, 同じ変数を使っていないようにする. その上で, \mathcal{G}_1 と \mathcal{G}_2 の両者の生成規則と合併するとともに, 新しい開始変数 S_0 を用意して, 生成規則 $S_0 \rightarrow S_1 \mid S_2$ を追加する.

$$\mathcal{G}: \quad S_0 \rightarrow S_1 \mid S_2$$
$$S_1 \rightarrow \dots$$
$$\dots\dots\dots$$
$$S_2 \rightarrow \dots$$
$$\dots\dots\dots$$

この CFG を \mathcal{G} とおくと,

$$L(\mathcal{G}) = L(\mathcal{G}_1) \cup L(\mathcal{G}_2)$$

が成立している.

連結演算に関する閉包性 2つの CFG \mathcal{G}_1 と \mathcal{G}_2 が与えられたとする.

$$\mathcal{G}_1: \quad S_1 \rightarrow \dots$$

$$\mathcal{G}_2: \quad S_2 \rightarrow \dots$$

このとき, 変数は適当に変更して, \mathcal{G}_1 と \mathcal{G}_2 では, 同じ変数を使っていないようにする. その上で, \mathcal{G}_1 と \mathcal{G}_2 の両者の生成規則と合併するとともに, 新しい開始変数 S_0 を用意して, 生成規則 $S_0 \rightarrow S_1 S_2$ を追加する.

$$\mathcal{G}: \quad S_0 \rightarrow S_1 S_2$$
$$S_1 \rightarrow \dots$$
$$\dots\dots\dots$$
$$S_2 \rightarrow \dots$$
$$\dots\dots\dots$$

この CFG を \mathcal{G} とおくと,

$$L(\mathcal{G}) = L(\mathcal{G}_1) \circ L(\mathcal{G}_2)$$

が成立している.

スター演算に関する閉包性 CFG \mathcal{G}_1 が与えられたとする.

$$\mathcal{G}_1: \quad S_1 \rightarrow \dots$$

このとき、新しい開始変数 S_0 を用意して、生成規則 $S_0 \rightarrow S_1 S_0$ および $S_0 \rightarrow \varepsilon$ を追加する。

$$\mathcal{G} : \begin{array}{l} S_0 \rightarrow S_1 S_0 \mid \varepsilon \\ S_1 \rightarrow \dots \\ \dots \end{array}$$

この CFG を \mathcal{G} とおくと、

$$L(\mathcal{G}) = L(\mathcal{G}_1)^*$$

が成立している。

積集合に関する閉包性がないこと 以下の2つの言語 L_1 と L_2 を考える。

$$\begin{aligned} L_1 &= \{a^n b^n c^m \mid n, m \geq 0\} \\ L_2 &= \{a^n b^m c^m \mid n, m \geq 0\} \end{aligned}$$

このとき、

$$\begin{aligned} \mathcal{G}_1 : \quad & S_1 \rightarrow T_1 U_1 \\ & T_1 \rightarrow a T_1 b \mid \varepsilon \\ & U_1 \rightarrow c U_1 \mid \varepsilon \\ \mathcal{G}_2 : \quad & S_2 \rightarrow U_2 T_2 \\ & T_2 \rightarrow b T_2 c \mid \varepsilon \\ & U_2 \rightarrow a U_2 \mid \varepsilon \end{aligned}$$

をとると、 $L(\mathcal{G}_1) = L_1$, $L(\mathcal{G}_2) = L_2$ となるので、 L_1 と L_2 は文脈自由言語となる。このとき、

$$L_1 \cap L_2 = \{a^n b^n c^n \mid n \geq 0\}$$

となるが、後々示されるように (例 2.36)、これは文脈自由言語でない。つまり、 L_1 と L_2 が文脈自由言語だからといって、 $L_1 \cap L_2$ が文脈自由言語とは限らない。

補集合に関する閉包性がないこと もし補集合に関する閉包性があるとする。このとき、和集合に関する閉包性を利用して、任意の文脈自由言語 L_1 と L_2 について、

$$L_1 \cap L_2 = \overline{\overline{L_1} \cap \overline{L_2}}$$

が、いつも文脈自由言語になることになってしまう。しかし、これは、積集合に関する閉包性がないことに矛盾する。

つまり、 L_1 が文脈自由言語だからといって、その補集合 $\overline{L_1}$ が文脈自由言語とは限らない。