

前回の復習

Σ : アルファベット。言語を構成する記号の集合

Σ^* : 有限の長さの記号全体の集合

言語: Σ^* の部分集合

句構造文法

$G = (N, T, P, S)$:

P : $\alpha \rightarrow \beta, \alpha \in (N \cup T)^+, \beta \in (N \cup T)^*$ という形の
書き換え規則の集合

N : 非終端記号の集合

T : 終端記号の集合

S : 開始記号の集合

N と T は一文字の記号からなる集合で、 ϵ や 01 を要素として含むことはない

$L(G)$: G の開始記号を書き換えて得られる終端記号列の集合。

正規文法

プログラムの中の符号無し二進整数：0 または (1 の後 $\{0, 1\}$ が 0 個以上続く続く文字列)

これを表す句構造文法

$$G = (\{D, U\}, \{0, 1\}, P, U),$$

$$P = \{U \rightarrow 0|1D, \quad D \rightarrow \epsilon|0D|1D\}$$

$$\text{例: } U \Longrightarrow 0$$

$$U \Longrightarrow 1D \Longrightarrow 1$$

$$U \Longrightarrow 1D \Longrightarrow 10$$

上の文法は生成規則が $A \rightarrow aB$ または $A \rightarrow a$ または $A \rightarrow \epsilon$ (但し A, B は非終端記号で、 a は終端記号) という特徴がある。

C 言語で関数名や変数名になり得る文字列も上のような形の言語で記述できる。このような言語を正規言語と呼び、この講義の前半で学ぶ。

正規文法の定義

授業で使う定義: 正規文法とは、句構造文法 $G = (N, T, P, S)$ で生成規則がすべて

$$A \rightarrow \epsilon,$$

$$A \rightarrow a,$$

$$A \rightarrow aB$$

のいずれかであるもの。但し、大文字は非終端記号で、小文字は終端記号。

教科書 42 ページの定義: 正規文法とは、句構造文法 $G = (N, T, P, S)$ で右辺に ϵ を含まない生成規則はすべて

$$A \rightarrow a,$$

$$A \rightarrow aB$$

のいずれかであり、生成規則が右辺に ϵ を含む規則は有るとすれば $S \rightarrow \epsilon$ だけであるもの。

正規言語: 正規文法により生成される言語

正規文法の定義 2

教科書の定義の方が制限が強いので、授業の定義で表せるが、教科書の定義で表せない言語がありそうに見える。しかし、授業の定義による文法から教科書の定義を満たす等価な文法を作ることができる。作り方は後で説明する。

正規文法はどのように使われるのか？

ある記号列がある正規文法により定義される言語に属するかどうかは、後で述べるオートマトンにより容易に判定できる。オートマトンは1文字ずつ記号列を読み、そこまで読んだ記号列が言語に属するかどうか判定する。そこで、プログラムを1文字ずつオートマトンに入力して言語に属するかどうか判定し、言語に属する記号列ごとに区切る。

$L_1 \cup L_2$ も正規言語

L_1, L_2 が正規言語であるとき $L_1 \cup L_2$ も正規言語である。

正規文法の記述を簡潔に行う正規表現をあとで学ぶが、上の事実から、正規表現が正規文法を表せることがわかる。

$L_1 = L(G_1), L_2 = L(G_2), G_1 = (N_1, T_1, P_1, S_1), G_2 = (N_2, T_2, P_2, S_2)$ とする。 $L_1 \cup L_2$ を生成する文法を作る

1. 非終端記号を置き換えて $N_1 \cap N_2 = \emptyset$ にする
2. $G = (N_1 \cup N_2 \cup \{S\}, T_1 \cup T_2, P, S)$ で生成規則の集合を

$$P = P_1 \cup P_2 \cup \{S \rightarrow \alpha_1 | \cdots | \alpha_m | \beta_1 | \cdots | \beta_n\}$$

とする。但し $S_1 \rightarrow \alpha_1 | \cdots | \alpha_m \in P_1, S_2 \rightarrow \beta_1 | \cdots | \beta_n \in P_2$.

例: $P_1 = \{S \rightarrow \epsilon | a | aS\}, P_2 = \{S \rightarrow \epsilon | b | bS\}$

1. 非終端記号を置き換えて $P'_1 = \{S_1 \rightarrow \epsilon | a | aS_1\}, P'_2 = \{S_2 \rightarrow \epsilon | b | bS_2\}$ とする
2. $P = P'_1 \cup P'_2 \cup \{S \rightarrow \epsilon | a | aS_1 | b | bS_2\}$
 $= \{S_1 \rightarrow \epsilon | a | aS_1, S_2 \rightarrow \epsilon | b | bS_2,$
 $S \rightarrow \epsilon | a | aS_1 | b | bS_2\}.$

L_1L_2 も正規言語

L_1, L_2 が正規言語であるとき L_1L_2 も正規言語である。

正規文法の記述を簡潔に行う正規表現をあとで学ぶが、上の事実から、正規表現が正規文法を表せることがわかる。

$L_1 = L(G_1), L_2 = L(G_2), G_1 = (N_1, T_1, P_1, S_1), G_2 = (N_2, T_2, P_2, S_2)$ とする。 L_1L_2 を生成する文法を作る

1. 非終端記号を置き換えて $N_1 \cap N_2 = \emptyset$ にする
2. P_1 の生成規則を以下のように置き換え、置き換えた生成規則の集合を P_1'' とする

置き換え前	置き換え後
$A \rightarrow a$	$A \rightarrow aS_2$
$A \rightarrow \epsilon$	$A \rightarrow (S_2 \text{ の右辺})$

3. 求める文法は $(N_1 \cup N_2, T_1 \cup T_2, P_1'' \cup P_2, S_1)$

L_1L_2 も正規言語 2

例: $P_1 = \{S \rightarrow \epsilon|a|aS\}$, $P_2 = \{S \rightarrow \epsilon|b|bS\}$

1. 非終端記号を置き換えて $P'_1 = \{S_1 \rightarrow \epsilon|a|aS_1\}$,
 $P'_2 = \{S_2 \rightarrow \epsilon|b|bS_2\}$ とする

2. P''_1 のための置き換え

置き換え前	置き換え後
$S_1 \rightarrow aS_1$	$S_1 \rightarrow aS_1$
$S_1 \rightarrow a$	$S_1 \rightarrow aS_2$
$S_1 \rightarrow \epsilon$	$S_1 \rightarrow \epsilon$
	$S_1 \rightarrow b$
	$S_1 \rightarrow bS_2$

3. 求める文法は $(\{S_1, S_2\}, \{a, b\}, P''_1 \cup P'_2, S_1)$.

L^* も正規言語

L が正規言語であるとき L^* も正規言語である。

$L = L(G)$, $G = (N, T, P, S)$ とする。 L^* を生成する文法を作るには、 $S \rightarrow \epsilon$ を P に追加する。また以下の表の右の規則を P に追加する（追加する代わりに置き換えてもよい）

P の規則	新たに追加する規則
$A \rightarrow a$	$A \rightarrow aS$
$A \rightarrow \epsilon$	$A \rightarrow (S \text{ の右辺})$

例: $P = \{S \rightarrow \epsilon | a | b\}$ 。このとき $L(G) = \{\epsilon, a, b\}$ 。

P の規則	新たに追加する規則
$S \rightarrow a$	$S \rightarrow aS$
$S \rightarrow b$	$S \rightarrow bS$
$S \rightarrow \epsilon$	$S \rightarrow a$
$S \rightarrow \epsilon$	$S \rightarrow b$

生成規則の重複を除いて、できる生成規則の集合は

$$P' = \{S \rightarrow \epsilon | a | b | aS | bS\}$$

授業の定義による正規文法を教科書の定義に変換する方法

ϵ 生成規則：右辺が ϵ である生成規則

授業の定義による正規文法を $G = (N, T, P, S)$ とし、教科書の定義による文法 $G' = (N', T', P', S')$ を以下のように作る。

1. $P' = \emptyset$ とする
2. P の ϵ 生成規則でない規則をそのまま P' に追加
3. ϵ 生成規則の左辺の非終端記号をすべて列挙 (これを ϵ 生成記号と呼ぶ)
4. $P \ni A \rightarrow aB$ について B が ϵ 生成記号ならば $A \rightarrow a$ を P' に追加
5. $S \rightarrow \epsilon \in P$ なら、 $S' \rightarrow \epsilon$ (P' の S の規則の右辺) を P' に追加し、 $N' = N \cup \{S'\}$ とする。 $S \rightarrow \epsilon \notin P$ なら、 $S' = S$, $N' = N$.

例: $P = \{S \rightarrow \epsilon | aS | bB, B \rightarrow \epsilon | bB\}$

2. $P' = \{S \rightarrow aS | bB, B \rightarrow bB\}$

3. S, B

4. $P' = \{S \rightarrow a | b | aS | bB, B \rightarrow b | bB\}$

5. $P' = \{S' \rightarrow \epsilon | a | b | aS | bB, S \rightarrow a | b | aS | bB, B \rightarrow b | bB\}$

演習問題一覽

答案に学籍番号、名前、ふりがなを書いて下さい。わからないことが有れば、他の学生と相談するか、TA・教員に質問して下さい。

演習問題 10 アルファベット $\Sigma = \{0, 1\}$ 上の言語 $L_1 = \{1, 10\}^*$ を生成する正規文法を書け

演習問題 11 アルファベット $\Sigma = \{0, 1\}$ 上の言語 $L_2 = \{01, 11\}$ を生成する正規文法を書け (注意: $S \rightarrow 01|11$ は正規文法ではない)

演習問題 12 L_2^* を生成する正規文法を問 11 の解答を用いて授業で説明したやり方に沿って作れ

演習問題 13 $L_1 L_2^*$ (L_1 と L_2^* の接続) を生成する正規文法を問 10,12 の解答を用いて授業で説明したやり方に沿って作れ

演習問題 14 $L_1 \cup L_2^*$ (L_1 と L_2^* の和集合) を生成する正規文法を問 10,12 の解答を用いて授業で説明したやり方に沿って作れ

演習問題 15 前回演習で間違いが有った場合、×にされた理由がわかったかどうか / 納得できたかどうか書いて下さい。今日の授業でわかりにくい所や要望を書いて下さい