計算理論

Thomas Zeugmann

北海道大学 大学院情報科学研究科
アルゴリズム研究室

http://www-alg.ist.hokudai.ac.jp/~thomas/ToC/

第 2 回：形式文法の導入
(日本語訳：湊 真一，大久保 好章)

文法

言語の生成が意味するものを定式化しよう．自然言語においては，次のような状況が考えられる．Σ は，その言語のすべての単語から成り，大きくはあるが有限である．通常，自然言語を話したり書いたりすることで，文が形成される．典型的な文は，名詞句で始まり，後ろに動詞句が続く．こうした生成は次のように記述できるであろう．

$$< \text{文} > \rightarrow < \text{名詞句} >< \text{動詞句} >$$

文法

言語の生成が意味するものを定式化しよう．自然言語においては，次のような状況が考えられる．Σ は，その言語のすべての単語から成り，大きくはあるが有限である．通常，自然言語を話したり書いたりすることで，文が形成される．典型的な文は，名詞句で始まり，後ろに動詞句が続く．こうした生成は次のように記述できるであろう．

$$< 文 > \rightarrow < 名詞句 >< 動詞句 >$$

明らかに，より複雑な文は，より複雑な規則によって生成される．しかし，例えば，英語のよくある文法書を見ても，文を生成する有限個の規則しか載っていない．

形式文法

このことは，次に示す文法の一般的な定義を示唆している．

定義 1

以下を満たす $\mathcal{G} = [T, N, \sigma, P]$ を**文法 *(grammar)*** と呼ぶ．

(1) T と N はアルファベットであり， $T \cap N = \emptyset$.

(2) $\sigma \in N$,

(3) $P \subseteq ((T \cup N)^+ \setminus T^*) \times (T \cup N)^*$ は有限集合である．

形式文法

このことは，次に示す文法の一般的な定義を示唆している．

定義 1

以下を満たす $\mathcal{G} = [T, N, \sigma, P]$ を**文法 (*grammar*)** と呼ぶ．

(1) T と N はアルファベットであり，$T \cap N = \emptyset$.

(2) $\sigma \in N$,

(3) $P \subseteq ((T \cup N)^+ \setminus T^*) \times (T \cup N)^*$ は有限集合である．

T を終端アルファベット (*terminal alphabet*)，N を非終端アルファベット (*nonterminal alphabet*) と呼ぶ．また，σ は開始記号 (*start symbol*) であり，P は**生成規則**・書き換え規則 (*productions*) の集合であり，これらは単に規則 (*rules*) とも呼ばれる．

形式文法

このことは，次に示す文法の一般的な定義を示唆している．

定義 1

以下を満たす $\mathcal{G} = [T, N, \sigma, P]$ を**文法 (*grammar*)** と呼ぶ．

(1) T と N はアルファベットであり，$T \cap N = \emptyset$.

(2) $\sigma \in N$,

(3) $P \subseteq ((T \cup N)^+ \setminus T^*) \times (T \cup N)^*$ は有限集合である．

T を終端アルファベット (*terminal alphabet*)，N を非終端アルファベット (*nonterminal alphabet*) と呼ぶ．また，σ は開始記号 (*start symbol*) であり，P は生成規則・書き換え規則 (*productions*) の集合であり，これらは単に規則 (*rules*) とも呼ばれる．

$\alpha \in (T \cup N)^+ \setminus T^*$ と $\beta \in (T \cup N)^*$ に対して，生成規則を $\alpha \to \beta$ と記述することが通例である．

文法による言語の生成 I

次に，文法を用いて言語を生成する方法を説明する．これは以下のように定義される．

定義 2

ある文法 $\mathcal{G} = [T, N, \sigma, P]$ において，$\alpha', \beta' \in (T \cup N)^*$ とする．もしも，ある $\alpha, \beta \in (T \cup N)^*$ について，$\alpha \rightarrow \beta$ という規則が P に含まれていて，なおかつ $\alpha' = \alpha_1 \alpha \alpha_2$, $\beta' = \alpha_1 \beta \alpha_2$ であるならば，α' は β' を直接生成する ***(directly generate)*** と言い，$\alpha' \Rightarrow \beta'$ と表す．また，関係 $\Rightarrow$ の反射的推移的閉包を $\overset{*}{\Rightarrow}$ と表す．

例

例. 1

文法$\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える．ここで，
$P = \{\sigma \to \lambda, \sigma \to a, \sigma \to b, \sigma \to a\sigma a, \sigma \to b\sigma b\}$.

$\sigma \to a$ が P 中にあるので，σ から a を直接生成できる．また，規則 $\sigma \to a\sigma a$，$\sigma \to b\sigma b$ および $\sigma \to \lambda$ を用いて，σ から文字列 $abba$ を生成できる．すなわち，以下を得る．

$$\sigma \Rightarrow a\sigma a \Rightarrow ab\sigma ba \Rightarrow abba\,. \tag{1}$$

式 (1) の系列は，生成 (*generation*) あるいは導出 (*derivation*) と呼ばれる．文字列 s が非終端記号 h から生成できるとき，$h \overset{*}{\Rightarrow} s$ と書く．

例

例. 1

文法$\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える．ここで，
$P = \{\sigma \to \lambda, \sigma \to a, \sigma \to b, \sigma \to a\sigma a, \sigma \to b\sigma b\}$.

$\sigma \to a$ が P 中にあるので，σ から a を直接生成できる．また，規則 $\sigma \to a\sigma a$，$\sigma \to b\sigma b$ および $\sigma \to \lambda$ を用いて，σ から文字列 $abba$ を生成できる．すなわち，以下を得る．

$$\sigma \Rightarrow a\sigma a \Rightarrow ab\sigma ba \Rightarrow abba\,. \tag{1}$$

式 (1) の系列は，生成 (*generation*) あるいは導出 (*derivation*) と呼ばれる．文字列 s が非終端記号 h から生成できるとき，$h \overset{*}{\Rightarrow} s$ と書く．

例

例. 1

文法$\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える．ここで，
$P = \{\sigma \to \lambda, \sigma \to a, \sigma \to b, \sigma \to a\sigma a, \sigma \to b\sigma b\}$.

$\sigma \to a$ が P 中にあるので，σ から a を直接生成できる．また，規則 $\sigma \to a\sigma a$，$\sigma \to b\sigma b$ および $\sigma \to \lambda$ を用いて，σ から文字列 $abba$ を生成できる．すなわち，以下を得る．

$$\sigma \Rightarrow a\sigma a \Rightarrow ab\sigma ba \Rightarrow abba\,. \tag{1}$$

式 (1) の系列は，生成 (*generation*) あるいは導出 (*derivation*) と呼ばれる．文字列 s が非終端記号 h から生成できるとき，$h \overset{*}{\Rightarrow} s$ と書く．

文法による言語の生成 II

最後に，文法により生成される言語を定義する．

文法による言語の生成 II

最後に，文法により生成される言語を定義する．

定義 3

$\mathcal{G} = [T, N, \sigma, P]$ を文法とする．$\mathcal{G}$ により生成される言語 ***(language)*** $L(\mathcal{G})$ は，次の通り定義される．

$L(\mathcal{G}) = \{s \mid s \in T^* \text{ かつ } \sigma \overset{*}{\Rightarrow} s\}$.

文法による言語の生成 II

最後に，文法により生成される言語を定義する．

定義 3

$\mathcal{G} = [T, N, \sigma, P]$ を文法とする．$\mathcal{G}$ により生成される言語 ***(language)*** $L(\mathcal{G})$ は，次の通り定義される．

$L(\mathcal{G}) = \{s \mid s \in T^* \text{ かつ } \sigma \overset{*}{\Rightarrow} s\}$.

定義 2を満たす文法により生成される言語の族を $\mathcal{L}_0$ と表す．これら言語は 0 型言語 (*type-0 languages*) とも呼ばれる．ここで，0 は制約なしであることを示す．

例 - 回文 I

回文 (*palindrome*) は，左から読んでも，右から読んでも同じ文字列であったことを思い出そう．例えば，

AKASAKA

例 - 回文 I

回文 (*palindrome*) は，左から読んでも，右から読んでも同じ文字列であったことを思い出そう．例えば，

AKASAKA

トビコミコビト
にわとりとことりとわに
テングノハハノグンテ

例 - 回文 I

回文 (*palindrome*) は，左から読んでも，右から読んでも同じ文字列であったことを思い出そう．例えば，

AKASAKA

トビコミコビト
にわとりとことりとわに
テングノハハノグンテ

ここで再び，$\Sigma = \{a, b\}$ 上のすべての回文から成る言語，すなわち，$L_{pal} = \{w \mid w \in \{a, b\}^*,\ w = w^T\}$ について考えよう．

例 - 回文 I

回文 (*palindrome*) は，左から読んでも，右から読んでも同じ文字列であったことを思い出そう．例えば，

AKASAKA

トビコミコビト
にわとりとことりとわに
テングノハハノグンテ

ここで再び，$\Sigma = \{a, b\}$ 上のすべての回文から成る言語，すなわち，$L_{pal} = \{w \mid w \in \{a, b\}^*, w = w^T\}$ について考えよう．

例 1の文法，つまり，$\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える．ここで，$P = \{\sigma \to \lambda, \sigma \to a, \sigma \to b, \sigma \to a\sigma a, \sigma \to b\sigma b\}$ である．

例 - 回文 II

$L_{pal} = L(\mathcal{G})$ を証明しよう.

例 - 回文 II

$L_{pal} = L(\mathcal{G})$ を証明しよう.

主張 1. $L_{pal} \subseteq L(\mathcal{G})$.

数学的帰納法により証明する. まず 帰納法の基底 (induction basis) として, $w = \lambda, w = a$, および $w = b$ の場合を考える. P は $\sigma \to \lambda, \sigma \to a, \sigma \to b$ を含むので, それぞれの場合について $\sigma \overset{*}{\Rightarrow} w$ が成り立つ.

例 - 回文 II

$L_{pal} = L(\mathcal{G})$ を証明しよう.

主張 1. $L_{pal} \subseteq L(\mathcal{G})$.

数学的帰納法により証明する. まず 帰納法の基底 (induction basis) として, $w = \lambda, w = a$, および $w = b$ の場合を考える. P は $\sigma \to \lambda, \sigma \to a, \sigma \to b$ を含むので, それぞれの場合について $\sigma \overset{*}{\Rightarrow} w$ が成り立つ.
帰納的ステップ (Induction Step): $|w| \geqslant 2$ とする. $w = w^T$ であるから, w の最初と最後は必ず同じ記号である. すなわち, $w = avb$ または $w = bvb$ であって, v もまた回文である.
仮定より $\sigma \overset{*}{\Rightarrow} v$ であるから,

$$\sigma \Rightarrow a\sigma a \overset{*}{\Rightarrow} avb \quad \text{よって } w = avb \text{ の場合が示された,}$$
$$\sigma \Rightarrow b\sigma b \overset{*}{\Rightarrow} bvb \quad \text{よって } w = bvb \text{ の場合が示された.}$$

以上で主張 1 が示された.

例 - 回文 III

主張 2. $L(\mathcal{G}) \subseteq L_{pal}$.
基底: もしも 1 ステップだけしか導出しないとしたら，右側に σ を含まない書き換え規則，すなわち $\sigma \to \lambda, \sigma \to a$, または $\sigma \to b$ しか適用できない．したがって，$\sigma \Rightarrow w$ から $w = \lambda$, $w = a$ または $w = b$ だけが得られるので， $w \in L_{pal}$ が成立する．

例 - 回文 III

主張 2. $L(\mathcal{G}) \subseteq L_{pal}$.

基底: もしも 1 ステップだけしか導出しないとしたら，右側に σ を含まない書き換え規則，すなわち $\sigma \to \lambda, \sigma \to a$, または $\sigma \to b$ しか適用できない．したがって，$\sigma \Rightarrow w$ から $w = \lambda$, $w = a$ または $w = b$ だけが得られるので， $w \in L_{pal}$ が成立する．

帰納的ステップ: $n+1$ 回 $(n \geqslant 1)$ の導出を行うと仮定すると，

$$\sigma \Rightarrow a\sigma a \overset{*}{\Rightarrow} a\nu a \quad \text{または}$$
$$\sigma \Rightarrow b\sigma b \overset{*}{\Rightarrow} b\nu b \quad \text{である．}$$

仮定より，$\nu \in L_{pal}$ であるから，どちらの場合でも $w \in L_{pal}$ が成立する． ∎

正規文法

定義 4

ある文法 $\mathcal{G} = [T, N, \sigma, P]$ において，P のすべての規則 $\alpha \rightarrow \beta$ で $\alpha \in N$ と $\beta \in T^* \cup T^*N$ を満たすとき，$\mathcal{G}$ は**正規文法である *(regular)***と言う．

ある言語 L において，$L = L(\mathcal{G})$ となる正規文法 $\mathcal{G}$ が存在するならば，L は **正規言語である *(regular)*** と言う．すべての正規言語の集合を $\mathcal{REG}$ で表す．

正規文法

定義 4

ある文法 $\mathcal{G} = [T, N, \sigma, P]$ において，P のすべての規則 $\alpha \to \beta$ で $\alpha \in N$ と $\beta \in T^* \cup T^*N$ を満たすとき，$\mathcal{G}$ は**正規文法である *(regular)***と言う．

ある言語 L において，$L = L(\mathcal{G})$ となる正規文法 $\mathcal{G}$ が存在するならば，L は **正規言語である *(regular)*** と言う．すべての正規言語の集合を $\mathcal{REG}$ で表す．

例． 2

$P = \{\sigma \to ab,\ \sigma \to a\sigma\}$ である文法 $\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える．
$\mathcal{G}$ は正規文法であり，$L(\mathcal{G}) = \{a^n b \mid n \geqslant 1\}$ である．

正規言語の例

例. 3

$P = \{\sigma \to \lambda,\ \sigma \to a\sigma,\ \sigma \to b\sigma\}$ である文法
$\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える.

$\mathcal{G}$ は正規文法であり，また，$L(\mathcal{G}) = \Sigma^*$ である.

よって，Σ^* は正規言語である.

正規言語の例

例. 3

$P = \{\sigma \to \lambda,\ \sigma \to a\sigma,\ \sigma \to b\sigma\}$ である文法 $\mathcal{G} = [\{a, b\}, \{\sigma\}, \sigma, P]$ を考える.

$\mathcal{G}$ は正規文法であり，また，$L(\mathcal{G}) = \Sigma^*$ である.

よって，Σ^* は正規言語である.

例. 4

Σ を任意のアルファベット，$X \subseteq \Sigma^*$ を任意の有限集合とする．このとき，$P = \{\sigma \to s \mid s \in X\}$ である文法 $\mathcal{G} = [\Sigma, \{\sigma\}, \sigma, P]$ について，$L(\mathcal{G}) = X$ である.

したがって，すべての 有限 (*finite*) 言語は正規言語である.

他に正規であるものは？

問

どの言語が正規言語か？

他に正規であるものは？

問

どの言語が正規言語か？

この問に答えるためには，まず閉包 (*closure*) の性質から始めないといけない．

閉包性

定理 1

正規言語は，集合和，直積，クリーニ閉包の演算に関して閉じている．

閉包性

定理 1

正規言語は，集合和，直積，クリーニ閉包の演算に関して閉じている．

証明： L_1, L_2 を任意の正規言語とする．L_1, L_2 は正規言語であるから，正規文法 $\mathcal{G}_1 = [T_1, N_1, \sigma_1, P_1]$ および $\mathcal{G}_2 = [T_2, N_2, \sigma_2, P_2]$ が存在し，$i = 1, 2$ について $L_i = L(\mathcal{G}_i)$ である．ここで適当に非終端記号の名前を書き換えることにより $N_1 \cap N_2 = \emptyset$ を仮定しても一般性は失われない．
まず集合和から始める．$L = L_1 \cup L_2$ が正規言語であることを示す．

閉包性

定理 1

正規言語は，集合和，直積，クリーニ閉包の演算に関して閉じている．

証明： L_1, L_2 を任意の正規言語とする．L_1, L_2 は正規言語であるから，正規文法 $\mathcal{G}_1 = [T_1, N_1, \sigma_1, P_1]$ および $\mathcal{G}_2 = [T_2, N_2, \sigma_2, P_2]$ が存在し，$i = 1, 2$ について $L_i = L(\mathcal{G}_i)$ である．ここで適当に非終端記号の名前を書き換えることにより $N_1 \cap N_2 = \emptyset$ を仮定しても一般性は失われない．
まず集合和から始める．$L = L_1 \cup L_2$ が正規言語であることを示す．いま，

$$\mathcal{G}_{union} = [T_1 \cup T_2, N_1 \cup N_2 \cup \{\sigma\}, \sigma, P_1 \cup P_2 \cup \{\sigma \rightarrow \sigma_1, \sigma \rightarrow \sigma_2\}]$$

を考える．その構造より，$\mathcal{G}_{union}$ は正規文法である．

集合和に関する閉包性

主張 1. $L = L(\mathcal{G}_{union})$.

文字列のすべての生成 (導出) は σ から始まる．よって，ふたつの可能性，すなわち，$\sigma \rightarrow \sigma_1$ および $\sigma \rightarrow \sigma_2$ を考えることができる．前者の場合，σ_1 から始まるすべての導出を続行することができ，これは L_1 のすべての文字列を生成する．後者の場合は，σ_2 から続行し，L_2 のすべての文字列が生成される．よって，$L_1 \cup L_2 \subseteq L$ である．

集合和に関する閉包性

主張 1. $L = L(\mathcal{G}_{union})$.

文字列のすべての生成 (導出) は σ から始まる．よって，ふたつの可能性，すなわち，$\sigma \rightarrow \sigma_1$ および $\sigma \rightarrow \sigma_2$ を考えることができる．前者の場合，σ_1 から始まるすべての導出を続行することができ，これは L_1 のすべての文字列を生成する．後者の場合は，σ_2 から続行し，L_2 のすべての文字列が生成される．よって，$L_1 \cup L_2 \subseteq L$ である．

一方，その構造から，$L \subseteq L_1 \cup L_2$ であり，故に $L = L_1 \cup L_2$ となる． ∎ (集合和)

直積に関する閉包性 I

L_1L_2 が正規言語であることを示す．まず最初に思いつくのは，先程と同様な構築方法である．すなわち，新しい開始規則 $\sigma \rightarrow \sigma_1\sigma_2$ を作ることである．

残念ながら，これは正規文法の規則ではない．もう少し慎重に考える必要がある．しかし基本的アイデアはよいので，この規則を，順次構築する方法に置き換えてやればよい．

直積に関する閉包性 I

L_1L_2 が正規言語であることを示す．まず最初に思いつくのは，先程と同様な構築方法である．すなわち，新しい開始規則 $\sigma \rightarrow \sigma_1\sigma_2$ を作ることである．

残念ながら，これは正規文法の規則ではない．もう少し慎重に考える必要がある．しかし基本的アイデアはよいので，この規則を，順次構築する方法に置き換えてやればよい．

その方法は簡単に書ける．$s_1 \in L_1, s_2 \in L_2$ とすると，s_1s_2 を生成できればよい．そこで，σ_1 から始めて $\sigma_1 \Rightarrow w_1 \Rightarrow w_2 \Rightarrow \cdots \Rightarrow s_1$ と導出していって，ここで導出を終了せずに，s_2 を生成するための導出を継続することを考える．したがって，右側に $s_1\sigma_2$ を導出する書き換え規則があればよい．このアイデアは以下のように形式的に記述できる．

文法 $\mathcal{G}_{prod} = [T_1 \cup T_2, N_1 \cup N_2, \sigma_1, P]$ を考える．ここで，

$$\begin{aligned} P \quad = \quad & P_1 \setminus \{h \to s \mid s \in T_1^*,\ h \in N_1\} \\ & \cup \{h \to s\sigma_2 \mid h \to s \in P_1,\ s \in T_1^*\} \cup P_2 \,. \end{aligned}$$

とする．その構造より，$\mathcal{G}_{prod}$ は 正規文法である．

文法 $\mathcal{G}_{prod} = [T_1 \cup T_2, N_1 \cup N_2, \sigma_1, P]$ を考える．ここで，

$$\begin{aligned} P \;=\; & P_1 \setminus \{h \to s \mid s \in T_1^*,\ h \in N_1\} \\ & \cup \{h \to s\sigma_2 \mid h \to s \in P_1,\ s \in T_1^*\} \cup P_2\,. \end{aligned}$$

とする．その構造より，$\mathcal{G}_{prod}$ は 正規文法である．

主張 2. $L(\mathcal{G}_{prod}) = L_1 L_2$.

明らかに $L(\mathcal{G}_{prod}) \subseteq L_1 L_2$ なので，$L_1 L_2 \subseteq L(\mathcal{G}_{prod})$ を示そう．
$s \in L_1 L_2$ とすると，$s = s_1 s_2$ となる $s_1 \in L_1$ と $s_2 \in L_2$ が存在する．$s_1 \in L_1$ であるから，$\sigma_1 \Rightarrow w_1 \Rightarrow \cdots \Rightarrow w_n \Rightarrow s_1$ という導出が $\mathcal{G}_1$ に存在する．よって，w_n はただ 1 つの非終端記号 h を持たなければならず，$w_n = wh$ となる．$w_n \Rightarrow s_1$ かつ $s_1 \in T_1^*$ であるから，$wh \Rightarrow ws = s_1$ となるような規則 $h \to s$, $s \in T_1^*$ が適用されなければならない．

文法 $\mathcal{G}_{prod} = [T_1 \cup T_2, N_1 \cup N_2, \sigma_1, P]$ を考える．ここで，

$$\begin{aligned} P \;=\; & P_1 \setminus \{h \to s \mid s \in T_1^*,\ h \in N_1\} \\ & \cup \{h \to s\sigma_2 \mid h \to s \in P_1,\ s \in T_1^*\} \cup P_2 \,. \end{aligned}$$

とする．その構造より，$\mathcal{G}_{prod}$ は 正規文法である．

主張 2. $L(\mathcal{G}_{prod}) = L_1 L_2$.

明らかに $L(\mathcal{G}_{prod}) \subseteq L_1 L_2$ なので，$L_1 L_2 \subseteq L(\mathcal{G}_{prod})$ を示そう．
$s \in L_1 L_2$ とすると，$s = s_1 s_2$ となる $s_1 \in L_1$ と $s_2 \in L_2$ が存在する．$s_1 \in L_1$ であるから，$\sigma_1 \Rightarrow w_1 \Rightarrow \cdots \Rightarrow w_n \Rightarrow s_1$ という導出が $\mathcal{G}_1$ に存在する．よって，w_n はただ 1 つの非終端記号 h を持たなければならず，$w_n = wh$ となる．$w_n \Rightarrow s_1$ かつ $s_1 \in T_1^*$ であるから，$wh \Rightarrow ws = s_1$ となるような規則 $h \to s$, $s \in T_1^*$ が適用されなければならない．
しかし $\mathcal{G}_{prod}$ では，それらすべての生成規則が $h \to s\sigma_2$ に置き換えられている．したがって，最後の導出 $w_n \Rightarrow s_1$ は $wh \Rightarrow ws\sigma_2$ に置き換えられる．そして，$s_2 \in L_2$ であるから，P_2 の書き換え規則を適用して s_2 を生成することができる． ∎ (直積)

クリーニ閉包に関する閉包性

ある正規言語 L と正規文法 $\mathcal{G} = [T, N, \sigma, P]$ において $L = L(\mathcal{G})$ であるとする. このとき, L^* が正規言語であることを証明する.

クリーニ閉包に関する閉包性

ある正規言語 L と正規文法 $\mathcal{G} = [T, N, \sigma, P]$ において $L = L(\mathcal{G})$ であるとする．このとき，L^* が正規言語であることを証明する．
定義より $L^* = \bigcup_{i \in \mathbb{N}} L^i$ である．$L^0 = \{\lambda\}$ であるから，λ が生成できることを確認しておく必要がある．もし $\lambda \in L$ であればこれは明らかである．そうでない場合，書き換え規則 $\sigma \to \lambda$ を単純に加えれば良い．以降は，直積演算の場合と同様の考え方で証明できる．すなわち，

$$\mathcal{G}^* = [T, N \cup \{\sigma^*\}, \sigma^*, P^*], \text{ ここで}$$

$$P^* = P \cup \{h \to s\sigma \mid h \to s \in P,\ s \in T^*\} \cup \{\sigma^* \to \sigma,\ \sigma^* \to \lambda\}$$

なる文法を考える．$L(\mathcal{G}^*) = L^*$ の証明は，演習問題とする． ∎

文法の等価性

最後に，文法の等価性を定義して，今回の講義を終える．

定義 5

$\mathcal{G}$ と $\hat{\mathcal{G}}$ を任意の文法とする．$L(\mathcal{G}) = L(\hat{\mathcal{G}})$ であるとき，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である *(equivalent)* と言う．

文法の等価性

最後に，文法の等価性を定義して，今回の講義を終える．

定義 5

$\mathcal{G}$ と $\hat{\mathcal{G}}$ を任意の文法とする．$L(\mathcal{G}) = L(\hat{\mathcal{G}})$ であるとき，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である ***(equivalent)*** と言う．

等価な文法の例として，
$\mathcal{G} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \rightarrow a\sigma a,\ \sigma \rightarrow aa,\ \sigma \rightarrow a\}]$,
および
$\hat{\mathcal{G}} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \rightarrow a,\ \sigma \rightarrow a\sigma\}]$. を考える．

文法の等価性

最後に，文法の等価性を定義して，今回の講義を終える．

定義 5

$\mathcal{G}$ と $\hat{\mathcal{G}}$ を任意の文法とする．$L(\mathcal{G}) = L(\hat{\mathcal{G}})$ であるとき，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である *(equivalent)* と言う．

等価な文法の例として，
$\mathcal{G} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \to a\sigma a,\ \sigma \to aa,\ \sigma \to a\}]$,
および
$\hat{\mathcal{G}} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \to a,\ \sigma \to a\sigma\}]$. を考える．

そうすると，$L(\mathcal{G}) = \{a\}^+ = L(\hat{\mathcal{G}})$ であることは容易にわかる．したがって，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である．

文法の等価性

最後に，文法の等価性を定義して，今回の講義を終える．

定義 5

$\mathcal{G}$ と $\hat{\mathcal{G}}$ を任意の文法とする．$L(\mathcal{G}) = L(\hat{\mathcal{G}})$ であるとき，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である *(equivalent)* と言う．

等価な文法の例として，
$\mathcal{G} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \to a\sigma a,\ \sigma \to aa,\ \sigma \to a\}]$,
および
$\hat{\mathcal{G}} = [\{a\}, \{\sigma\}, \sigma, \{\sigma \to a,\ \sigma \to a\sigma\}]$. を考える．

そうすると，$L(\mathcal{G}) = \{a\}^+ = L(\hat{\mathcal{G}})$ であることは容易にわかる．したがって，$\mathcal{G}$ と $\hat{\mathcal{G}}$ は等価である．

ただし，$\hat{\mathcal{G}}$ は正規文法であるが，$\mathcal{G}$ はそうでないことに注意しよう．

Thank you!