オートマトンと言語理論

山本真基

２０２０年９月

オートマトンと言語理論の基礎を学習する．オートマトンとは，計算の原理を解明するために考案された数学的モデルである．言語理論とは，プログラミング言語の（文法に関する）数学的モデルである形式言語を扱う理論分野である．オートマトンと形式言語は，それぞれ異なった分野で考案されたモデルであるが，それらの間には密接な関係がある．ここでは，言語とは何か？，から始め，オートマトンと形式言語，それにそれらの間の関係を学習する．

目次

第 1 章

導入

オートマトンとは何か，言語理論とは何か，について簡単に説明する．まず，双方で扱われる「言語」とは何か，について説明する．

1.1　言語とは

定義 1.1

「文字の集合」を**アルファベット**という．Σ をアルファベットとしたとき，Σ 上の**文字列**とは，（重複を許して）Σ の文字を一列に並べたものである．

x を Σ 上の文字列とする．x をなす文字の個数（種類ではない）を x の**長さ**といい，$|x|$ と表す．長さ 0 の文字列を**空列**といい，ϵ と表す．

以降では，アルファベット Σ の大きさは有限とする．（つまり，$|\Sigma| \neq \infty$.）

注 1.1. ここでいうアルファベットという用語は，a, b, c,. . . , z, A, B, C,. . . , Z といった，英文を構成する，いわゆる「アルファベット」とは異なる．また，空列と空白文字とを混同しないこと．（空列は長さ 0 の「文字列」であり，空白文字はある一つの「文字」である．）

例 1.1 (アルファベット)**.** 以下はアルファベットの例である．

1. $\Sigma = \{0, 1\}$
2. $\Sigma = \{0, 1, 2, \ldots, 9, \mathrm{a,b,c,\ldots,z,A,B,C,\ldots,Z}, \#, \$, \%\}$
3. $\Sigma = \{0, 1, (,)\}$

例 1.2 (文字列)**.** $\Sigma = \{0, 1, (,)\}$ をアルファベットとする．以下は Σ 上の文字列である．

1. 0101001
2. (101)(((0
3. ((000)(111))(010)(101)

一方，00#11 は Σ 上の文字列でない．（$\# \notin \Sigma$ なので．）

問 1.1. アルファベットを $\Sigma = \{\mathrm{a,b,c}\}$ とする．Σ 上の文字列とそうでないものをそれぞれ一つずつあげなさい．

事実 1.1. アルファベット Σ をキーボードから入力できる半角の文字の集合とする．p をＣ＋＋言語の（全角文字のない）あるソースコードとする．このとき，p は Σ 上の文字列である．

例 1.3. 図 1.1 のＣ＋＋言語のソースコードを考える．このソースコードは，（改行を空

```
#include<iostream>
using namespace std;

int main()
{
        int a=1;
        cout << a << endl;
}
```

図 1.1: Ｃ＋＋言語のソースコード

白文字とみなした場合）以下の文字列となる．アルファベット Σ をキーボードから入力

#include<iostream> using namespace std; int main() { int a=1; cout << a << endl; }

図 1.2: ソースコードを横一列に記述したもの

できる半角の文字の集合としたとき，これは Σ 上のある文字列である．もちろん，コンパイルエラー及び実行エラーをおこすソースコードも Σ 上の文字列である．

定義 1.2

Σ をアルファベットとする．Σ 上の文字列の集合を Σ 上の**言語**という．

L を言語とする．L の要素の個数を L の**大きさ**といい，$|L|$ と表す．大きさ 0 の言語は空集合であり，$\emptyset$ と表す．

例 1.4 (言語と大きさ)**.** アルファベットを $\Sigma = \{\mathrm{a,b,c}\}$ とする．以下の集合 L は Σ 上のある言語である．

$$L \;=\; \{\mathrm{a}, \mathrm{ab}, \mathrm{abc}, \mathrm{ababa}, \mathrm{abbcca}\}.$$

このとき，$|L| = 5$ である．

例 1.5 (言語と大きさ)**.** アルファベットを $\Sigma = \{0, 1\}$ とする．以下の集合 L は Σ 上のある言語である．

$$L \;=\; \{x : \exists n \in \mathbb{N}[x \text{ は } n \text{ の 2 進表記}]\}.$$

このとき，$|L|$ は有限でない．

例 1.6 (言語と大きさ)**.** アルファベット Σ をキーボードから入力できる半角の文字の集合とする．以下の集合 L は Σ 上のある言語である．

$$L \;=\; \{p : p \text{ はコンパイルエラーのないC＋＋言語のソースコード }\}.$$

このとき，$|L|$ は有限でない．

問 1.2. Σ をアルファベットとする．Σ（それ自体）は Σ 上の言語となりえるか．

定義 1.3

Σ をアルファベットとする．k 個の文字 $x_1, x_2, \ldots, x_k \in \Sigma$ を（左から右へ）一列に並べた「文字列」を $x_1 x_2 \cdots x_k$ と表記する．$x_1 = x_2 = \cdots = x_k = \mathrm{a} \in \Sigma$ であるとき，$x_1 x_2 \cdots x_k$ を a^k と表記する．$k = 0$ のとき，a^0 は空列 ϵ を意味するものとする．

定義 1.4

Σ をアルファベットとする．$\Sigma \circ \Sigma$ を Σ と Σ の**連接**といい，以下のように定義する．

$$\Sigma \circ \Sigma \;\stackrel{\text{def}}{=}\; \{xy : x, y \in \Sigma\}.$$

一般に，任意の $k \in \mathbb{N}$ について，

$$\Sigma^k \stackrel{\text{def}}{=} \underbrace{\Sigma \circ \Sigma \circ \cdots \circ \Sigma}_{k\ 個} \stackrel{\text{def}}{=} \{x_1 x_2 \ldots x_k : x_1, x_2, \ldots, x_k \in \Sigma\}$$
$$\Sigma^0 \stackrel{\text{def}}{=} \{\epsilon\}$$

Σ からなるすべての文字列の集合を Σ^* と表記する．つまり，

$$\Sigma^* \stackrel{\text{def}}{=} \bigcup_{k \in \mathbb{N} \cup \{0\}} \Sigma^k.$$

注 1.2. Σ をアルファベットとする．x が Σ 上の文字列であるとは，$x \in \Sigma^*$ であることである．また，L が Σ 上の言語であるとは，$L \subseteq \Sigma^*$ であることである．

例 1.7. アルファベットを $\Sigma = \{0, 1\}$ とする．このとき，

1. $\Sigma \circ \Sigma = \{00, 01, 10, 11\}$
2. $\Sigma^3 = \{000, 001, 010, 011, 100, 101, 110, 111\}$

よって，

1. Σ 上のある文字列 $x =$000 について，$x \in \Sigma^3 \subseteq \Sigma^*$.
2. Σ 上のある言語 $L = \{000, 111\}$ について，$L \subseteq \Sigma^3 \subseteq \Sigma^*$.

命題 1.1. Σ をアルファベットとする．$|\Sigma| = n$ のとき，$|\Sigma^k| = n^k$ である．（よって，$k = 0$ のとき，$|\Sigma^0| = |\{\epsilon\}| = 1$.）

問 1.3. 上の命題が成り立つ理由を説明しなさい．

問 1.4. Σ をアルファベットとする．（$\Sigma \neq \emptyset$ とする．）$\Sigma \circ \Sigma^*$ はどのような言語か説明しなさい．（Σ^* とは異なる！）

問 1.5. 以下が成り立つ理由を説明しなさい．

1. $\epsilon \in \Sigma^*$

2. $\emptyset^* = \{\epsilon\}$

1.2　言語理論とは

アルファベットを $\Sigma = \{0, 1, 2, \ldots, 9, .\}$ とする．Σ 上の以下の言語 L を考える．

$$L \stackrel{\text{def}}{=} \{a \in \Sigma^* : a \text{ は0以上の実数の１０進表記}\}.$$

例えば，

$$\begin{array}{lrcl} 1. & 0 & \in & L \\ 2. & 012 & \notin & L \\ 3. & 3.1415926 & \in & L \\ 4. & .123 & \notin & L \\ 5. & -1 & \notin & L \\ 6. & 12345.000 & \in & L \\ 7. & 10/3 & \notin & L \end{array}$$

この言語 L を「定義する記述方法」（の一つ）は以下である．$\Sigma' = \Sigma \setminus \{.\}$, $\Sigma'' = \Sigma' \setminus \{0\}$ として，

$$\{0\} \ \cup \ \Sigma'' \circ \Sigma'^* \ \cup \ (\{0\} \cup \Sigma'' \circ \Sigma'^*) \circ \{.\} \circ \Sigma' \circ \Sigma'^*$$

このように，言語を「定義する記述方法」や「生成する規則」を考えていくのが言語理論である．

1.3　オートマトンとは

アルファベットを $\Sigma = \{0, 1, 2, \ldots, 9, .\}$ とする．Σ 上の以下の言語 L を考える．

$$L \stackrel{\text{def}}{=} \{a \in \Sigma^* : a \text{ は0以上の実数の１０進表記}\}.$$

この言語 L を「識別する機械」は以下である．

このように，言語を「識別する機械」を考えていくのがオートマトンの理論である．

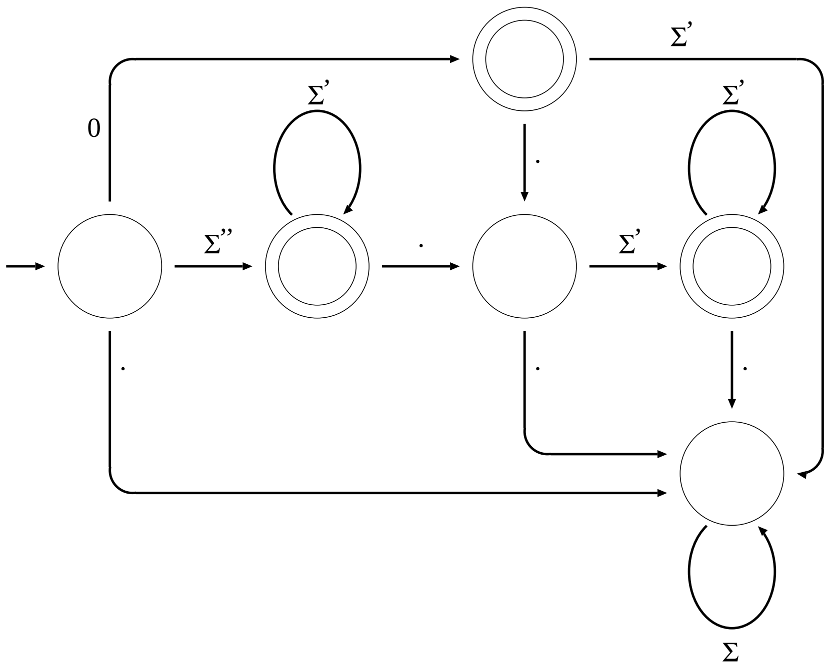


図 1.3: L を識別するオートマトン

第 2 章

正規言語

正規言語と呼ばれるある言語のクラスを考え，正規言語を定義する記述方法を考える．

2.1 正規表現

定義 2.1

Σ をアルファベットとする．L を Σ 上の言語する．k 個の文字列 $x_1, x_2, \ldots, x_k \in L$ を（左から右へ）一列に並べた「文字列」を $x_1 x_2 \cdots x_k$ と表記する．$x_1 = x_2 = \cdots = x_k = y \in L$ であるとき，$x_1 x_2 \cdots x_k$ を y^k と表記する．$k = 0$ のとき，y^0 は空列 ϵ を意味するものとする．

定義 2.2

以下の三つの演算を**正規演算**という．Σ をアルファベット，L, L_1, L_2 を Σ 上の任意の言語としたとき，

$$\begin{array}{lcll} \text{和} & : & L_1 \cup L_2 & \stackrel{\text{def}}{=} \{x : x \in L_1 \text{ または } x \in L_2\}, \\ \text{連接} & : & L_1 \circ L_2 & \stackrel{\text{def}}{=} \{xy : x \in L_1 \text{ かつ } y \in L_2\}, \\ \text{スター} & : & L^* & \stackrel{\text{def}}{=} \{x_1 x_2 \ldots x_k : k \in \mathbb{N} \cup \{0\} \text{ かつ } x_1, x_2, \ldots, x_k \in L\}. \end{array}$$

例 2.1 (正規演算，和)**.** $\Sigma = \{0, 1, \#\}$ をアルファベット，

$$\begin{array}{rcl} L_1 & = & \{0^n \# 1^n : n \in \mathbb{N} \cup \{0\}\}, \\ L_2 & = & \{1^n \# 0^n : n \in \mathbb{N} \cup \{0\}\}, \end{array}$$

を Σ 上の言語とする．このとき，

$$L_1 \cup L_2 \; = \; \{0^n \# 1^n, 1^n \# 0^n : n \in \mathbb{N} \cup \{0\}\}.$$

問 2.1. 以下が成り立つ理由を説明しなさい．

1. $L \cup \emptyset = L$
2. $\epsilon \notin L$ のとき，$L \cup \{\epsilon\} \neq L$

例 2.2 (正規演算，連接)**.** $\Sigma = \{0, 1, \#\}$ をアルファベット，

$$\begin{array}{rcl} L_1 & = & \{0, 1\}, \\ L_2 & = & \{\epsilon, \#0, \#1\}, \end{array}$$

を Σ 上の言語とする．このとき，

$$\begin{array}{rcl} L_1 \circ L_2 & = & \{0, 1, 0\#0, 0\#1, 1\#0, 1\#1\} \\ L_2 \circ L_1 & = & \{0, 1, \#00, \#01, \#10, \#11\} \end{array}$$

問 2.2. 以下が成り立つ理由を説明しなさい．

1. $L \circ \{\epsilon\} = \{\epsilon\} \circ L = L$
2. $L \circ \emptyset = \emptyset \circ L = \emptyset$

例 2.3 (正規演算，スター)**.** $\Sigma = \{0, 1, \#\}$ をアルファベット，

$$L = \{00\#, 11\#\},$$

を Σ 上の言語とする．このとき，

$$L^* = \{\epsilon, 00\#, 11\#, 00\#00\#, 00\#11\#, 11\#00\#, 11\#11\#, 00\#00\#00\#, \ldots\}.$$

問 2.3. $\Sigma = \{0, 1\}$ をアルファベット，$L = \{00, 11\}$ とする．L^* を求めなさい．このとき，000 を部分文字列として含む文字列が L^* の要素になることはあるか．また，010 や 101 や 111 などはどうか．

注 2.1. 演算記号が混在した表記では，演算の結合の順序は，$*$, $\circ$, $\cup$ の順とする．例えば，$L_1 \circ L_2 \cup L^*$ は，$(L_1 \circ L_2) \cup (L^*)$ を表す．また，演算の結合を優先したい場合は，かっこ記号を用いて優先度を明示する．例えば，先の例にかっこ記号をつけて $(L_1 \circ (L_2 \cup L))^*$ とすれば（かっこ記号がないものとは）異なる言語になる．

定義 2.3

Σ をアルファベットとする．Σ 上の**正規言語**（**正則言語**）L を，以下のように帰納的に定義する．

1. $\emptyset, \{\epsilon\}$ は正規言語である．
2. 任意の $x \in \Sigma$ に対して，$\{x\}$ は正規言語である．
3. 任意の正規言語 L, L_1, L_2 に対して，$L_1 \cup L_2$, $L_1 \circ L_2$, L^* は正規言語である．

Σ 上の正規言語を，Σ 及びその要素，正規演算 $\cup, \circ, *$ を用いて（上の再帰的な構成に従って）表した表記を**正規表現**（**正則表現**）という．

R を正規表現とする．R が表す言語（正規言語）を $L(R)$ と表記する．正規言語のクラス（正規言語の全集合）を $\mathcal{L}_{\mathrm{REG}}$ と表記する．

注 2.2. 有限オートマトン（次章で学習する）が受理する言語を正規言語と定義する教科書もある．

例 2.4 (正規言語とその正規表現)**.** アルファベットを $\Sigma = \{0, 1, \#\}$ とする．

1. $L = \{0, 1\}$ は正規言語である．その正規表現は，$\{0\} \cup \{1\}$ である．
2. $L = \{00, 11, 0\#, 1\#, 01\#, 10\#\}$ は正規言語である．その正規表現は，

$$\{0\}\circ\{0\}\cup\{1\}\circ\{1\}\cup\{0\}\circ\{\#\}\cup\{1\}\circ\{\#\}\cup\{0\}\circ\{1\}\circ\{\#\}\cup\{1\}\circ\{0\}\circ\{\#\}.$$

3. $L = \{w \in \Sigma^* : w$ は 1 を少なくとも一つ含む $\}$ は正規言語である．その正規表現は，

$$\Sigma^* \circ \{1\} \circ \Sigma^*.$$

命題 2.1. Σ をアルファベットとする．（$|\Sigma|$ は有限．）任意の $L \subseteq \Sigma$ は，Σ 上の正規言語である．更に，すべての要素が有限長である（有限の）言語は正規言語である．

問 2.4. 上の命題が成り立つ理由を説明しなさい．

以降，すべての要素が有限長である（有限の）言語は，集合そのものを正規表現とみなす．例えば，例 2.4 の 2 において，言語 L そのものを正規表現とみなす．

例 2.5 (正規表現とその言語)**.** $\Sigma = \{0, 1\}$ をアルファベットとする．以下は，正規表現 R とそれが表す正規言語 $L(R)$ である．

正規表現 R	正規言語 $L(R)$
$\{0\}^* \circ \{1\} \circ \{0\}^*$	$\{w \in \Sigma^* : w$ は 1 をちょうど一つ含む $\}$
$\{0\}^* \circ \{1\} \circ \{0\}^* \circ \{1\} \circ \{0\}^*$	$\{w \in \Sigma^* : w$ は 1 をちょうど二つ含む $\}$
$\Sigma^* \circ \{010\} \circ \Sigma^*$	$\{w \in \Sigma^* : w$ は文字列 010 を含む $\}$
$\Sigma^* \circ \{010\}$	$\{w \in \Sigma^* : w$ は文字列 010 で終わる $\}$
$\{0\} \circ \Sigma^* \cup \Sigma^* \circ \{1\}$	$\{w \in \Sigma^* : w$ は 0 で始まるか 1 で終わる $\}$

命題 2.2. アルファベットを $\Sigma = \{0, 1\}$ とする．$\{0^n : n \in \mathbb{N}\}$，$\{1^n : n \in \mathbb{N}\}$，$\{0^i 1^j : i, j \in \mathbb{N}\}$ はいずれも正規言語である．一方，$\{0^n 1^n : n \in \mathbb{N}\}$ は正規言語でない．

証明. $\{0^n : n \in \mathbb{N}\}$，$\{1^n : n \in \mathbb{N}\}$，$\{0^i 1^j : i, j \in \mathbb{N}\}$ が正規言語であることは明らか.

問 2.5. その理由を説明しなさい.

$\{0^n 1^n : n \in \mathbb{N}\}$ が正規言語でないことは，次章で示される（ポンプの補題より導かれる）命題 3.7 を参照. ■

問 2.6. $\Sigma = \{0, 1\}$ をアルファベットとする．以下の正規表現 R が表す正規言語 $L(R)$ を求めなさい.

1. $\{0\} \circ \Sigma^* \circ \{0\} \cup \{1\} \circ \Sigma^* \circ \{1\}$.
2. $(\Sigma \circ \Sigma)^*$.
3. $\{1\}^* \circ (\{0\} \circ \{1\}^* \circ \{0\} \circ \{1\}^*)^*$.

問 2.7. $\Sigma = \{0, 1\}$ をアルファベットとする．以下の正規言語に対応する正規表現を求めなさい.

1. $\{w \in \Sigma^* : w$ は最初と最後の文字が異なる文字列 $\}$.
2. $\{w \in \Sigma^* : w$ は奇数長の文字列 $\}$.
3. $\{w \in \Sigma^* : w$ は奇数個の 0 を含む $\}$.

以上の例や問でみてきた正規表現や正規言語は，それらが対応していることが容易に分

かるものであった．一方，以下の命題が示すように，ある言語が正規言語であること，または，その正規表現が与えられたとしても，それらが対応するものであるかは明らかでないものがある．

命題 2.3. アルファベットを $\Sigma = \{0,1\}$ とする．言語 $L = \{w \in \Sigma^* :$ w は 00 を含まない文字列 $\}$ は正規言語である．

証明. 言語 L の正規表現が $\{1\}^* \circ (\{01\} \circ \{1\}^*)^* \circ \{\epsilon, 0\}$ であることを示す．つまり，その正規表現を R と表記すれば，$L(R) = L$ を示す．（$L(R) \subseteq L$ かつ $L(R) \supseteq L$ を示す．）

(⊆) $w \in L(R)$ を任意の文字列とする．（$w \in L$ を示せばよい．）以下，w が 0 で始まる場合を考える．（そうでない場合も同様にして示される．）まず，R の構成要素である $\{01\} \circ \{1\}^*$ が，言語 $\{011^n : n \in \mathbb{N} \cup \{0\}\}$ であることは明らかである．

問 2.8. 明らかである理由を説明しなさい．

その言語を A とおく．（つまり，$A = \{011^n : n \in \mathbb{N} \cup \{0\}\}$ とする．）このとき，（w が 0 で始まることから）$y_1, y_2, \ldots, y_k \in A$（$k \in \mathbb{N} \cup \{0\}$）が存在して，$w$ は以下のうちどちらか一方である．

$$\begin{array}{rcl} w &=& y_1 y_2 \ldots y_k \\ w &=& y_1 y_2 \ldots y_k 0 \end{array}$$

問 2.9. そのようになる理由を説明しなさい．

一つ目の場合を考える．（二つ目の場合も同様にして示される．）w に 00 が含まれないことを，言語 A からの文字列の個数 k についての帰納法により示す．$k = 0$ のとき，$w = \epsilon$ より明らか．任意の文字列 $y_1 y_2 \ldots y_{k-1}$（$y_i \in A$）に 00 が含まれないと仮定する．このとき，任意の文字列 $y_1 y_2 \ldots y_{k-1} y_k$（$y_i \in A$）を考える．$w' = y_1 y_2 \ldots y_{k-1}$ に 00 が含まれないことは帰納仮定より示される．また，w' は ϵ か 1 で終わる文字列のどちらかであり，更に，ある $n \in \mathbb{N} \cup \{0\}$ について $y_k = 011^n$ である．よって，$w = w' y_k$ に 00 が含まれることはない．

(⊇) $w \in L$ を任意の文字列とする．（$w \in L(R)$ を示せばよい．）w に表れる 0 の左側に $|$ を入れる．このように $|$ で区切られた w の文字列を左側から順に，$y_1, y_2, \ldots, y_k$（$k \in \mathbb{N} \cup \{0\}$）とおく．よって，$w = y_1 y_2 \ldots y_k$ である．

問 2.10. 例えば，$w = 111011101010$ であったとき，どのようになるか.

このとき，任意の $i \in [k]$ について，

$$\begin{array}{lll} i = 1 & : & y_i \in \{1\}^* \cup \{01\} \circ \{1\}^* \\ i \in [k] \setminus \{1, k\} & : & y_i \in \{01\} \circ \{1\}^* \\ i = k & : & \{01\} \circ \{1\}^* \cup \{\epsilon, 0\} \end{array}$$

問 2.11. この事実が成り立つ理由を，先の例 $w = 111011101010$ を用いて説明しなさい．また，$w = 010101110111$ であった場合はどうか.

よって，$w \in L(R)$ となる． ■

問 2.12. アルファベットを $\Sigma = \{0, 1\}$ とする．言語 $L = \{w \in \Sigma^* :$ w は 000 を含まない文字列 $\}$ の正規表現を示しなさい.

L を Σ 上の任意の正規言語とする．この章では，L を定義する記述方法，つまり，正規表現を考えた．では，Σ 上の任意の文字列 $x \in \Sigma^*$ が与えられたとき，文字列 x が言語 L に属するかどうか，つまり，$x \in L$ かどうかを判定するにはどうしたらよいだろうか？

章末問題

以下の問いに答えなさい．

1. $\Sigma = \{0,1\}$ をアルファベットとする．以下の言語に対応する正規表現を示しなさい．
   (a) $L = \{w \in \Sigma^* : w$ は 01 を含まない $\}$.
   (b) $L = \{w \in \Sigma^* : w$ は 10 を含まない $\}$.
   (c) $L = \{w \in \Sigma^* : w$ は 11 を含まない $\}$.
   (d) $L = \{w \in \Sigma^* : w$ は 001 を含まない $\}$.
   (e) $L = \{w \in \Sigma^* : w$ は 100 を含まない $\}$.
2. $\Sigma = \{0,1\}$ をアルファベットとする．以下の言語に対応する正規表現を示しなさい．
   (a) $L = \{w \in \Sigma^* : w$ は 0 も 1 も含まない $\}$.
   (b) $L = \{w \in \Sigma^* : w$ は 00 も 11 も含まない $\}$.
   (c) $L = \{w \in \Sigma^* : w$ は 000 も 111 も含まない $\}$.
3. $\Sigma = \{0,1\}$ をアルファベットとする．以下の正規表現 R が表す正規言語 $L(R)$ を求めなさい．
   (a) $\{\epsilon, 1\} \circ \{01\}^* \circ \{\epsilon, 0\}$.
   (b) $\{\epsilon, 0\} \circ (\{1\} \circ \{1\}^* \circ \{0\})^* \circ \{1\}^*$.
   (c) $\{\epsilon, 0, 00\} \circ (\{1\} \circ \{1\}^* \circ \{0, 00\})^* \circ \{1\}^*$.

第 3 章

有限オートマトン

正規言語を「識別」する「機械」を考える．

3.1　決定性有限オートマトン（DFA）

定義 3.1

決定性有限オートマトン（**DFA**）とは，以下で定義される「５つ組」$(Q, \Sigma, \delta, q_0, F)$ である．

$$\begin{array}{lcl} Q & : & \text{状態集合} \\ \Sigma & : & \text{入力アルファベット} \\ \delta & : & Q \times \Sigma \text{ から } Q \text{ への遷移関数} \\ q_0 & : & \text{開始状態（} q_0 \in Q \text{）} \\ F & : & \text{受理状態の集合（} F \subseteq Q \text{）} \end{array}$$

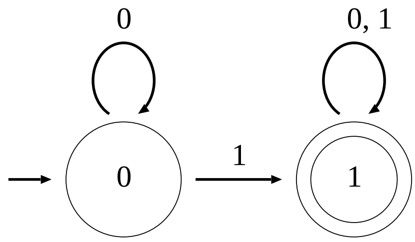
集合 Q, Σ, F はいずれも有限集合である．

例 3.1 (決定性有限オートマトン)**.** 決定性有限オートマトン $M_1 = (Q, \Sigma, \delta, q_0, \{q_1\})$ を次のように定義する．$Q = \{q_0, q_1\}$, $\Sigma = \{0, 1\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

	0	1
q_0	q_0	q_1
q_1	q_1	q_1

これを図示すると以下のようになる．

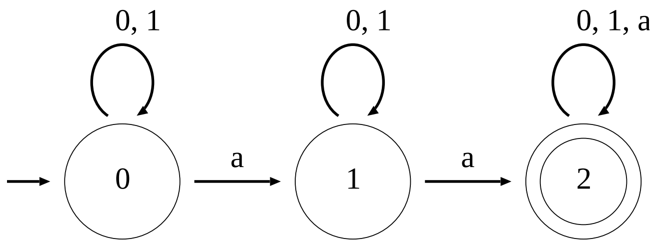
注 3.1. 遷移関数を表した図において，状態は○印で表される．そのうち，受理状態は◎印で表される．



例 3.2 (決定性有限オートマトン)**.** 決定性有限オートマトン $M_2 = (Q, \Sigma, \delta, q_0, \{q_2\})$ を次のように定義する．$Q = \{q_0, q_1, q_2\}$, $\Sigma = \{0, 1, \mathrm{a}\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

	0	1	a
q_0	q_0	q_0	q_1
q_1	q_1	q_1	q_2
q_2	q_2	q_2	q_2

これを図示すると以下のようになる．



定義 3.2

$M = (Q, \Sigma, \delta, q_0, F)$ を決定性有限オートマトンとする．$w = w_1 w_2 \ldots w_n$ を Σ 上の長さ n の文字列とする．（つまり，$w \in \Sigma^n$.）M が w を**受理する**とは，次を満たすことである．状態の列 $r_0, r_1, \ldots, r_n \in Q$ が存在して，

1. $r_0 = q_0$.
2. $\forall i \in [n][\delta(r_{i-1}, w_i) = r_i]$.
3. $r_n \in F$.

M に対して，$L(M) = \{w \in \Sigma^* : M$ が w を受理する $\}$ とする．このとき，M が言語 $L(M)$ を**認識する**という．

決定性有限オートマトンが認識する言語のクラスを $\mathcal{L}_{\mathrm{DFA}}$ と表記する．（つまり，$\mathcal{L}_{\mathrm{DFA}} = \{L :$ ある DFA が L を認識する $\}$.）

例 3.3 (認識する言語)**.** 例 3.1 の決定性有限オートマトンを M_1 とする．このとき，

$$\begin{array}{lcl} 1,\ 01,\ 10,\ 11,\ 00100 & \in & L(M_1) \\ 0,\ 00,\ 000,\ 00000 & \notin & L(M_1) \end{array}$$

つまり，M_1 が認識する言語 $L_1 = L(M_1)$ は，

$$L_1 \ = \ \{w \in \{0,1\}^* : w \text{ は 1 を少なくとも一つ含む }\}.$$

例 3.4 (認識する言語)**.** 例 3.2 の決定性有限オートマトンを M_2 とする．このとき，

$$\begin{array}{lcl} \text{aaa, 01a01a, a0a1a0a1a} & \in & L(M_2) \\ \text{0a1, 1a0, a01010} & \notin & L(M_2) \end{array}$$

つまり，M_2 が認識する言語 $L_2 = L(M_2)$ は，

$$L_2 \ = \ \{w \in \{0,1,a\}^* : w \text{ は二つ以上の a からなる }\}.$$

問 3.1. アルファベットを $\Sigma = \{0,1\}$ とする．以下の決定性有限オートマトンが認識する言語をそれぞれ求めなさい．（DFA を図示すると分かりやすくなる.）

1. $M = (Q, \Sigma, \delta, q_0, \{q_1\})$. $Q = \{q_0, q_1, q_2\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

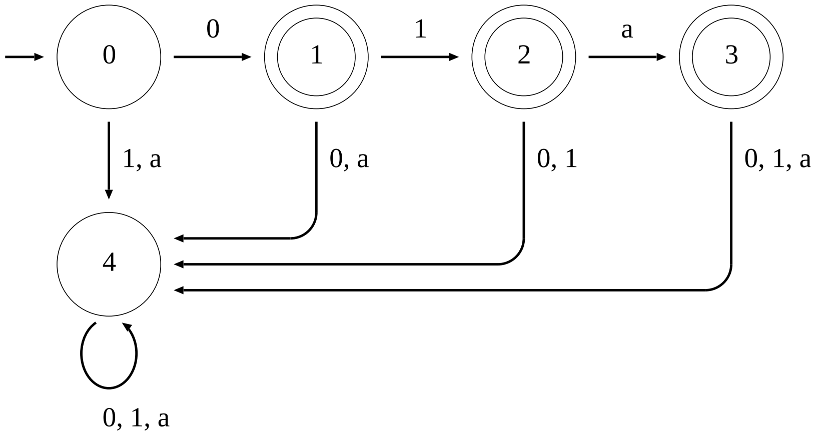
	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_2

2. $M = (Q, \Sigma, \delta, q_0, \{q_0\})$. $Q = \{q_0, q_1\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

	0	1
q_0	q_0	q_1
q_1	q_1	q_0

3. $M = (Q, \Sigma, \delta, q_0, \{q_0\})$. $Q = \{q_0, q_1, q_2\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

	0	1
q_0	q_0	q_1
q_1	q_1	q_2
q_2	q_2	q_0



例 3.5. アルファベットを $\Sigma = \{0, 1, \mathrm{a}\}$ とする．$L = \{0, 01, 01\mathrm{a}\}$ とする．この言語を認識する決定性有限オートマトンを図示すると以下のようになる．

これは次のように定義される．$M = (Q, \Sigma, \delta, q_0, F)$．$Q = \{q_0, q_1, q_2, q_3, q_4\}$，$\Sigma = \{0, 1, \mathrm{a}\}$，$F = \{q_1, q_2, q_3\}$ として，遷移関数 $\delta : Q \times \Sigma \to Q$ を以下のように定義する．

	0	1	a
q_0	q_1	q_4	q_4
q_1	q_4	q_2	q_4
q_2	q_4	q_4	q_3
q_3	q_4	q_4	q_4
q_4	q_4	q_4	q_4

問 3.2. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の言語を認識する決定性有限オートマトンをそれぞれ求めなさい．

1. $L = \{w \in \Sigma^* : w$ は文字列 010 を含む $\}$.
2. $L = \{w \in \Sigma^* : w$ は文字列 101 を含む $\}$.
3. $L = \{w \in \Sigma^* : w$ は文字列 010 で終わる $\}$.
4. $L = \{w \in \Sigma^* : w$ は文字列 101 で終わる $\}$.
5. $L = \{w \in \Sigma^* : w$ は偶数個の 1 と偶数個の 0 からなる $\}$.

3.2　非決定性有限オートマトン（NFA）

定義 3.3

非決定性有限オートマトン（**NFA**）とは，以下で定義される「5 つ組」$(Q, \Sigma, \delta, q_0, F)$ である．

$$
\begin{array}{lcl}
Q & : & \text{状態集合} \\
\Sigma & : & \text{入力アルファベット} \\
\delta & : & Q \times \Sigma \text{ から } 2^Q \text{ への遷移関数} \\
q_0 & : & \text{開始状態（}q_0 \in Q\text{）} \\
F & : & \text{受理状態の集合（}F \subseteq Q\text{）}
\end{array}
$$

集合 Q, Σ, F はいずれも有限集合である．

注 3.2. $Q = \{q_0, q_1, q_2\}$ であった場合，

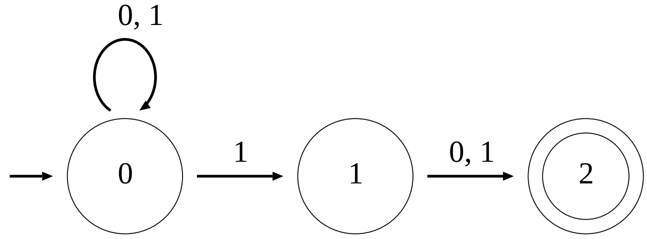
$$2^Q = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$$

となる．つまり，遷移先が状態集合 Q の部分集合である．これは，（決定性の DFA と異なって）遷移先が複数ある場合もあることを意味する．また，遷移先がない場合もあり，$\emptyset$ と表記する．

例 3.6 (非決定性有限オートマトン)**.** 非決定性有限オートマトン $N = (Q, \Sigma, \delta, q_0, \{q_2\})$ を次のように定義する．$Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}$ として，遷移関数 $\delta : Q \times \Sigma \to 2^Q$ を以下のように定義する．

	0	1
q_0	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\emptyset$	$\emptyset$

これを図示すると以下のようになる．



以降では，この例のように，遷移先が $\emptyset$ の場合，その遷移は図中で省略される．（この例では，$\delta(q_2, 0)$, $\delta(q_2, 1)$ が省略されている．）

問 3.3. 上の例において，どの部分が非決定的か説明しなさい．

定義 3.4

$N = (Q, \Sigma, \delta, q_0, F)$ を非決定性有限オートマトンとする．$w = w_1 w_2 \dots w_n$ を Σ 上の長さ n の文字列とする．（つまり，$w \in \Sigma^n$．）N が w を**受理する**とは，次を満たすことである．状態の列 $r_0, r_1, \dots, r_n \in Q$ が存在して，

1. $r_0 = q_0$.
2. $\forall i \in [n][r_i \in \delta(r_{i-1}, w_i)]$.
3. $r_n \in F$.

N に対して，$L(N) = \{w \in \Sigma^* : N$ が w を受理する $\}$ とする．このとき，N が言語 $L(N)$ を**認識する**という．

非決定性有限オートマトンが認識する言語のクラスを $\mathcal{L}_{\mathrm{NFA}}$ と表記する．（つまり，$\mathcal{L}_{\mathrm{NFA}} = \{L :$ ある NFA が L を認識する $\}$.）

注 3.3. 遷移先が $\emptyset$ である場合，（読み込まれていない入力文字列がまだあったとしても）その遷移の時点で受理されないことを意味する．

例 3.7 (認識する言語)**.** 例 3.6 の非決定性有限オートマトンを N とする．このとき，

$$\begin{array}{lcl} 10,\ 11,\ 111,\ 00010 & \in & L(N) \\ 0,\ 1,\ 00,\ 01,\ 101,\ 11101 & \notin & L(N) \end{array}$$

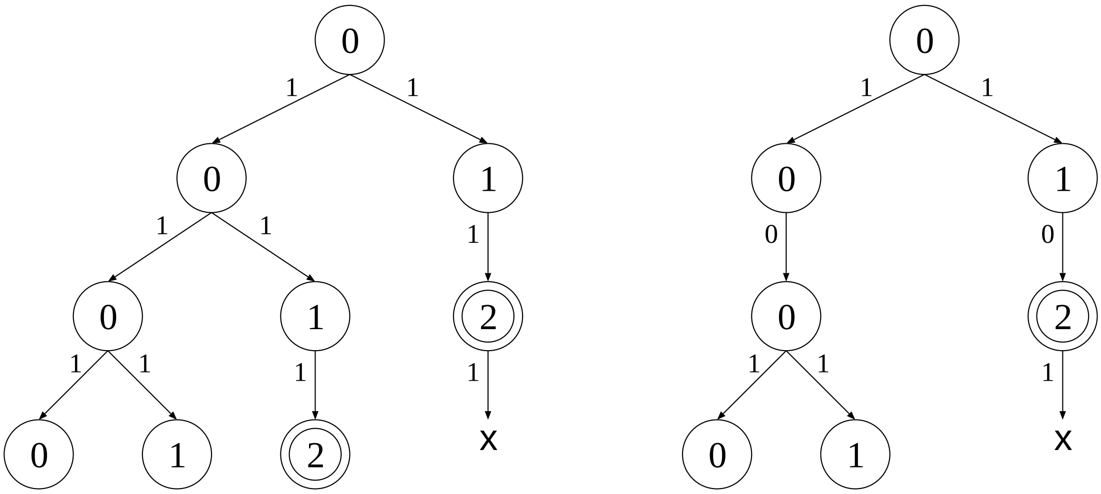
つまり，N が認識する言語 $L = L(N)$ は，

$$L \ = \ \{w \in \Sigma^* : w \text{ は最後から二つ目が } 1 \text{ である文字列}\}.$$

以下は，111 を受理する（左図），101 を受理しない（右図），非決定性の計算過程を示した図である．

問 3.4. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の非決定性有限オートマトンが認識する言語をそれぞれ求めなさい．（NFA を図示すると分かりやすくなる．）

1. $N = (Q, \Sigma, \delta, q_0, \{q_2\})$．$Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}$ として，遷移関数 $\delta : Q \times \Sigma \to 2^Q$ を以下のように定義する．



(a) 111 を受理する計算過程　(b) 101 を受理しない計算過程

図 3.1: 非決定性の計算過程

	0	1
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_1\}$	$\{q_1, q_2\}$
q_2	$\emptyset$	$\emptyset$

2. $N = (Q, \Sigma, \delta, q_0, \{q_2\})$．$Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}$ として，遷移関数 $\delta : Q \times \Sigma \to 2^Q$ を以下のように定義する．

	0	1
q_0	$\emptyset$	$\{q_1\}$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
q_2	$\emptyset$	$\emptyset$

問 3.5. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の言語を認識する非決定性有限オートマトンをそれぞれ求めなさい．

1. $L = \{w \in \Sigma^* : w$ は文字列 010 を含む $\}$.
2. $L = \{w \in \Sigma^* : w$ は文字列 010 で終わる $\}$.

問 3.6. 例 3.7 で挙げた言語 L を認識する DFA を求めることは可能か．可能であればその DFA を求めなさい．

DFA と NFA の等価性

定理 3.1. $\mathcal{L}_{\mathrm{NFA}} = \mathcal{L}_{\mathrm{DFA}}$.

定義 3.5

$M = (Q, \Sigma, \delta, q_0, F)$ を決定性有限オートマトンとする．遷移関数 δ に対して，$\delta^* : Q \times \Sigma^* \to Q$ を以下のように（文字列の長さについて）帰納的に定義する．任意の状態 $q \in Q$，任意の文字列 $s \in \Sigma^*$ について，

$$\delta^*(q, s) \;=\; \begin{cases} q & s = \epsilon \\ \delta(\delta^*(q, t), c) & s = tc \quad (c \in \Sigma) \end{cases}$$

$N = (Q, \Sigma, \delta, q_0, F)$ を非決定性有限オートマトンとする．遷移関数 δ に対して，$\delta^* : 2^Q \times \Sigma^* \to 2^Q$ を以下のように（文字列の長さについて）帰納的に定義する．任意の状態の集合 $q \subseteq Q$，任意の文字列 $s \in \Sigma^*$ について，

$$\delta^*(q, s) \;=\; \begin{cases} q & s = \epsilon \\ \displaystyle\bigcup_{q' \in \delta^*(q,t)} \delta(q', c) & s = tc \quad (c \in \Sigma) \end{cases}$$

例 3.8. 例 3.1 の DFA を M とする．このとき，

$$\begin{array}{rcl} \delta^*(q_0, 000) & = & q_0 \\ \delta^*(q_0, 010) & = & q_1 \\ \delta^*(q_1, 000) & = & q_1 \end{array}$$

となる．

例 3.9. 例 3.6 の NFA を N とする．このとき，

$$\begin{array}{rcl} \delta^*(\{q_0\}, 010) & = & \{q_0, q_2\} \\ \delta^*(\{q_0, q_1\}, 010) & = & \{q_0, q_2\} \\ \delta^*(\{q_1, q_2\}, 010) & = & \emptyset \end{array}$$

となる．

証明． $\mathcal{L}_{\mathrm{NFA}} \supseteq \mathcal{L}_{\mathrm{DFA}}$ かつ $\mathcal{L}_{\mathrm{NFA}} \subseteq \mathcal{L}_{\mathrm{DFA}}$ を示す．$\mathcal{L}_{\mathrm{NFA}} \supseteq \mathcal{L}_{\mathrm{DFA}}$ は明らか．

問 3.7. この事実（$\mathcal{L}_{\mathrm{NFA}} \supseteq \mathcal{L}_{\mathrm{DFA}}$）が成り立つ理由を説明しなさい．

以下，$\mathcal{L}_{\mathrm{NFA}} \subseteq \mathcal{L}_{\mathrm{DFA}}$ を示す．このためには，任意の非決定性有限オートマトン $N = (Q, \Sigma, \delta, q_0, F)$ に対して，ある決定性有限オートマトン $D = (Q_D, \Sigma_D, \delta_D, q_D, F_D)$ が N を模倣する（つまり，$L(D) = L(N)$）ことを示せばよい．N から D への変換方法を以下に示す．

- $Q_D = 2^Q$
- $\Sigma_D = \Sigma$
- $S \in Q_D$, $a \in \Sigma_D$ について，$\delta_D(S, a) = \bigcup_{q \in S} \delta(q, a)$
- $q_D = \{q_0\}$
- $F_D = \{S \in Q_D : S \cap F \neq \emptyset\}$

以下，$L(D) = L(N)$ であることを示す．そのためには，任意の $x \in \Sigma^*$ について $\delta_D^*(q_D, x) = \delta^*(\{q_0\}, x)$ を示せばよい．

問 3.8. その理由を説明しなさい．

これを $|x|$ についての帰納法により示す．まず，$|x| = 0$ のときは明らか．

問 3.9. 明らかである内容を説明しなさい．

$x \in \Sigma^*$ を任意に固定する．（$|x| \geq 1$.）$x = ya$ とする．（$y \in \Sigma^*$, $a \in \Sigma$.）以下の等式変形より示される．

$$\begin{aligned}
\delta_D^*(q_D, x) &= \delta_D^*(q_D, ya) \\
&= \delta_D(\delta_D^*(q_D, y), a) \quad (\because\ \delta_D^* \text{ の定義}) \\
&= \delta_D(\delta^*(\{q_0\}, y), a) \quad (\because\ \text{帰納仮定}) \\
&= \bigcup_{q \in \delta^*(\{q_0\}, y)} \delta(q, a) \quad (\because\ \delta_D \text{ の定義}) \\
&= \delta^*(q_0, ya) \quad (\because\ \delta^* \text{ の定義}) \\
&= \delta^*(q_0, x).
\end{aligned}$$

■

例 3.10. 例 3.6 の NFA $N = (Q, \Sigma, \delta, q_0, F)$ を，定理で示された変換方法を用いて，DFA $M = (Q_D, \Sigma_D, \delta_D, q_D, F_D)$ にすると，次のようになる．

- $Q_D = \{\emptyset, \{q_0\}, \{q_1\}, \{q_2\}, \{q_0, q_1\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$
- $\Sigma_D = \Sigma$
- $q_D = \{q_0\}$
- $F_D = \{\{q_2\}, \{q_0, q_2\}, \{q_1, q_2\}, \{q_0, q_1, q_2\}\}$

としたとき，遷移関数 δ_D は以下のようになる．

	0	1
$\emptyset$	$\emptyset$	$\emptyset$
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
$\{q_2\}$	$\emptyset$	$\emptyset$
$\{q_0, q_1\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$
$\{q_0, q_2\}$	$\{q_0\}$	$\{q_0, q_1\}$
$\{q_1, q_2\}$	$\{q_2\}$	$\{q_2\}$
$\{q_0, q_1, q_2\}$	$\{q_0, q_2\}$	$\{q_0, q_1, q_2\}$

問 3.10. 以下の言語を認識する NFA を求めなさい．それをもとに，上の定理で示された変換方法を用いて，その言語を認識する DFA を求めなさい．

1. $L = \{w00 : w \in \{0,1\}^*\}$.
2. $L = \{w11 : w \in \{0,1\}^*\}$.

3.3 ϵ-非決定性有限オートマトン（ϵ-NFA）

定義 3.6

Σ をアルファベットとする．ϵ を空列とする．（つまり，$|\epsilon| = 0$）$\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ とする．

定義 3.7

ϵ-非決定性有限オートマトン（ϵ-NFA）とは，以下で定義される「5 つ組」

$(Q, \Sigma, \delta, q_0, F)$ である．

$$\begin{array}{lcl} Q & : & \text{状態集合} \\ \Sigma & : & \text{入力アルファベット} \\ \delta & : & Q \times \Sigma_\epsilon \text{ から } 2^Q \text{ への遷移関数} \\ q_0 & : & \text{開始状態（}q_0 \in Q\text{）} \\ F & : & \text{受理状態の集合（}F \subseteq Q\text{）} \end{array}$$

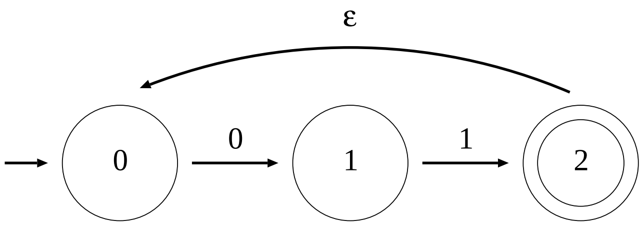
集合 Q, Σ, F はいずれも有限集合である．

ϵ-非決定性有限オートマトンが認識する言語のクラスを $\mathcal{L}_{\epsilon\text{-NFA}}$ と表記する．（つまり，$\mathcal{L}_{\epsilon\text{-NFA}} = \{L : \text{ある NFA が } L \text{ を認識する}\}$.）

例 3.11 (ϵ-非決定性有限オートマトン)**.** ϵ-非決定性有限オートマトン $N = (Q, \Sigma, \delta, q_0, \{q_2\})$ を次のように定義する．$Q = \{q_0, q_1, q_2\}, \Sigma = \{0, 1\}$ として，遷移関数 $\delta : Q \times \Sigma_\epsilon \to 2^Q$ を以下のように定義する．

	ϵ	0	1
q_0	$\{q_0\}$	$\{q_1\}$	$\emptyset$
q_1	$\{q_1\}$	$\emptyset$	$\{q_2\}$
q_2	$\{q_0, q_2\}$	$\emptyset$	$\emptyset$

これを図示すると以下のようになる．



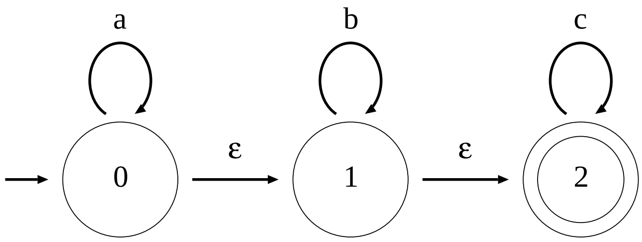
このとき，N が認識する言語 $L = L(N)$ は，

$$L = \{(01)^n : n \in \mathbb{N}\}.$$

注 3.4. 任意の $q \in Q$ について，$q \in \delta(q, \epsilon)$．上の例において，$\delta(q_2, \epsilon) = \{q_0, q_2\}$ による遷移を ϵ-**遷移**と呼ぶ．ϵ-遷移も非決定性の遷移の一つである．

問 3.11. 上の例で挙げた言語 L を認識する NFA を求めることは可能か．可能であればその NFA を求めなさい．

例 3.12 (ϵ-非決定性有限オートマトン)**.** $\Sigma = \{\mathrm{a}, \mathrm{b}, \mathrm{c}\}$ として，$L = \{\mathrm{a}^i\mathrm{b}^j\mathrm{c}^k \in \Sigma^* : i, j, k \in \mathbb{N} \cup \{0\}\}$ とする．L を認識する ϵ-非決定性有限オートマトン $N = (Q, \Sigma, \delta, q_0, \{q_2\})$ は次のようである．$Q = \{q_0, q_1, q_2\}$ として，遷移関数 $\delta : Q \times \Sigma_\epsilon \to 2^Q$ を以下とする．



問 3.12. 上の例で挙げた言語 L を認識する NFA を求めることは可能か．可能であればその NFA を求めなさい．

定理 3.2. $\mathcal{L}_{\epsilon\text{-NFA}} = \mathcal{L}_{\text{NFA}}$.

定義 3.8

$N = (Q, \Sigma, \delta, q_0, F)$ を ϵ-非決定性有限オートマトンとする．任意の状態 $q \in Q$ に対して，ϵ-遷移だけで遷移できる状態の集合を q の ϵ-**閉包**といい，ϵ-CL(q) と表す．

例 3.13. 例 3.11 の ϵ-NFA N について，$\epsilon\text{-CL}(q_2) = \{q_0, q_2\}$ となる．その他の状態 q については $\epsilon\text{-CL}(q) = \{q\}$．例 3.12 の ϵ-NFA N について，$\epsilon\text{-CL}(q_0) = \{q_0, q_1, q_2\}$, $\epsilon\text{-CL}(q_1) = \{q_1, q_2\}$, $\epsilon\text{-CL}(q_2) = \{q_2\}$ となる．

$\mathcal{L}_{\epsilon\text{-NFA}} \subseteq \mathcal{L}_{\text{NFA}}$ の証明の概略を示す．($\mathcal{L}_{\epsilon\text{-NFA}} \supseteq \mathcal{L}_{\text{NFA}}$ は明らか．) $\mathcal{L}_{\epsilon\text{-NFA}} \subseteq \mathcal{L}_{\text{NFA}}$ を示すためには，任意の ϵ-非決定性有限オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ に対して，ある非決定性有限オートマトン $N = (Q_N, \Sigma_N, \delta_N, q_N, F_N)$ が M を模倣する（つまり，$L(N) = L(M)$）ことを示せばよい．M から N への変換方法を以下に示す．

- $Q_N = Q$
- $\Sigma_N = \Sigma$
- $q \in Q_N$, $a \in \Sigma_N$ について，$\delta_N(q, a) \overset{\text{def}}{=} \epsilon\text{-CL}\left(\bigcup_{q' \in \epsilon\text{-CL}(q)} \delta(q', a)\right)$

- $q_N = q_0$
- $\epsilon\text{-CL}(q_0) \cap F = \emptyset$ なら $F_N = F$，そうでないなら $F_N = F \cup \{q_0\}$.

問 3.13. 例 3.11，例 3.12 それぞれについて，上記の変換方法を適用させなさい．

3.4　正規言語との等価性

定理 3.3. $\mathcal{L}_{\text{REG}} = \mathcal{L}_{\text{DFA}}$.

これは，（決定性）有限オートマトンが認識する言語は正規言語であることを言っている．更に，正規言語にはそれを認識する有限オートマトンが存在することを言っている．この定理を示すためには，$\mathcal{L}_{\text{REG}} \subseteq \mathcal{L}_{\text{DFA}}$ かつ $\mathcal{L}_{\text{REG}} \supseteq \mathcal{L}_{\text{DFA}}$ を示せばよい．

補題 3.4 ($\mathcal{L}_{\text{REG}} \subseteq \mathcal{L}_{\text{DFA}}$)**.** L を Σ 上の任意の正規言語とする．L を認識する有限オートマトンが存在する．

証明. 補題を示すためには，（定理 3.1，定理 3.2 より $\mathcal{L}_{\epsilon\text{-NFA}} = \mathcal{L}_{\text{DFA}}$ であることから）L を認識する ϵ-NFA を構成すればよい．正規言語が帰納的に定義される（定義 2.3）ことから，それぞれの帰納的定義について ϵ-NFA の構成を示せばよい．

【初期段階】正規表現の定義より，以下の三つの場合がある．

1. $L = \emptyset$
2. $L = \{\epsilon\}$
3. ある $x \in \Sigma$ に対して $L = \{x\}$

問 3.14. 上の三つの場合について，それぞれ NFA を図示しなさい．（実際には DFA となる．）

【帰納仮定】$k \in \mathbb{N} \cup \{0\}$ を任意とする．$\cup$, $\circ$, $*$ の適用が k 回以下の任意の正規言語について補題が成立している．

【帰納段階】L を，$\cup$, $\circ$, $*$ の適用が $k+1$ 回の正規言語とする．

1. $L = L_1 \cup L_2$ のとき．ただし，L_1, L_2 は，$\cup, \circ, *$ の適用が k 回以下の正規言語とする．帰納仮定より，L_1, L_2 を認識する NFA N_1, N_2 が存在する．L を認識する NFA N は，N_1, N_2 を用いて図 3.2 のようになる．

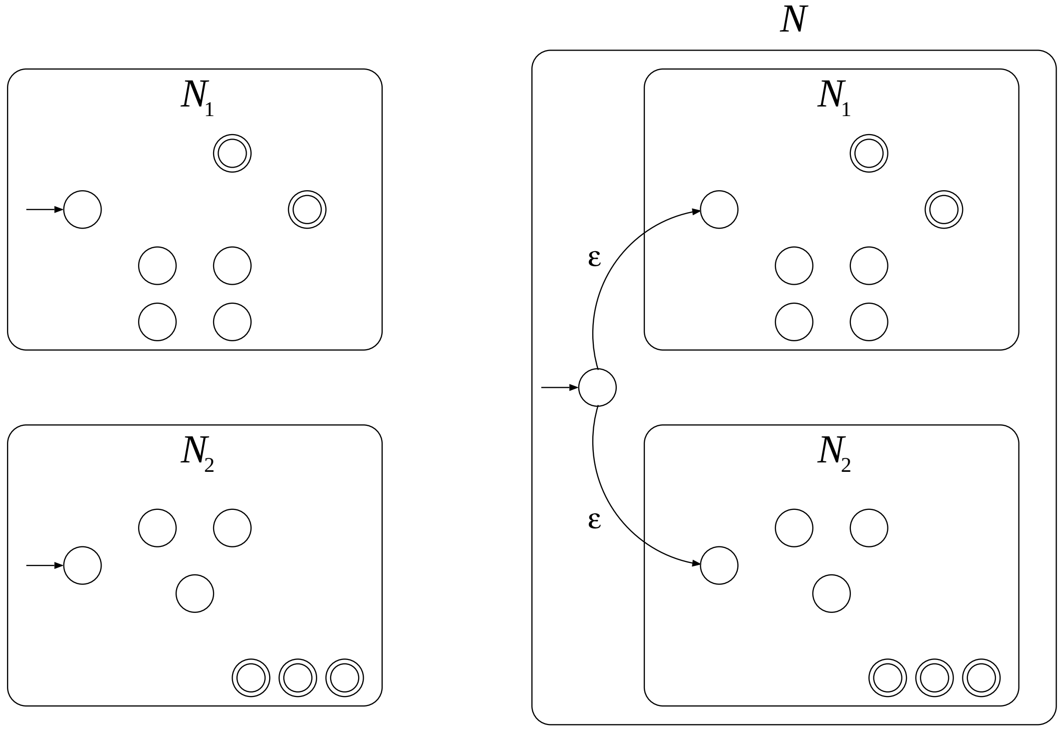


図 3.2: $L = L_1 \cup L_2$ のとき

2. $L = L_1 \circ L_2$ のとき．ただし，L_1, L_2 は，$\cup, \circ, *$ の適用が k 回以下の正規言語とする．帰納仮定より，L_1, L_2 を認識する NFA N_1, N_2 が存在する．L を認識する NFA N は，N_1, N_2 を用いて図 3.3 のようになる．
3. $L = L_0^*$ のとき．ただし，L_0 は，$\cup, \circ, *$ の適用が k 回以下の正規言語とする．帰納仮定より，L_0 を認識する NFA N_0 が存在する．L を認識する NFA N は，N_0 を用いて図 3.4 のようになる．

問 3.15. 上の三つの場合について，図示されたそれぞれの NFA を求めなさい．

以上より，L を認識する ϵ-NFA が存在する．（よって，L を認識する DFA が存在する．） ■

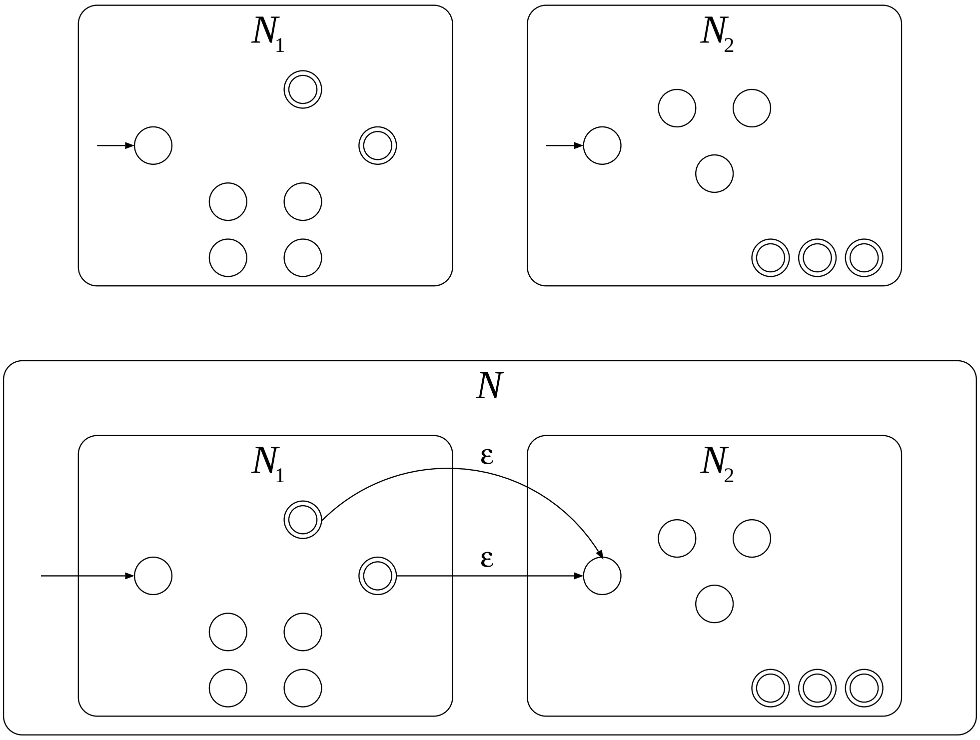


図 3.3: $L = L_1 \circ L_2$ のとき

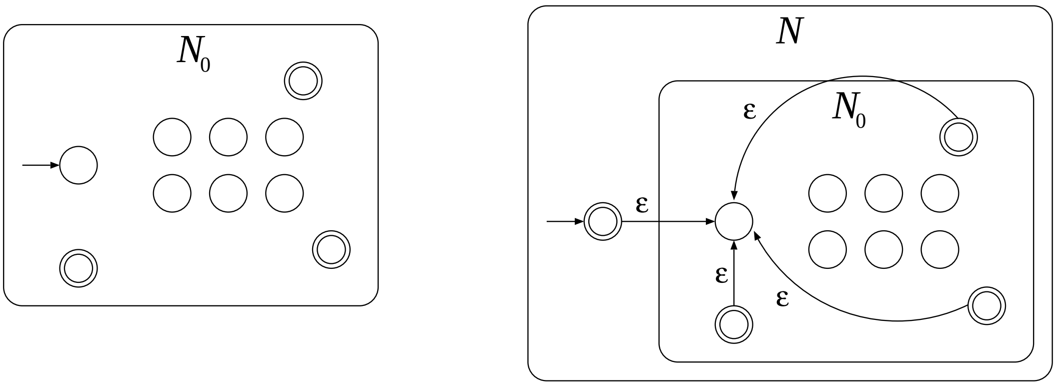


図 3.4: $L = L_0^*$ のとき

補題 3.5 ($\mathcal{L}_{\mathrm{REG}} \supseteq \mathcal{L}_{\mathrm{DFA}}$)**.** M を任意の有限オートマトン，L を M が認識する言語とする．L は正規言語である．

証明. 補題を示すためには，L を表す正規表現を示せばよい．このために，以下のような有限オートマトンを導入する．

定義 3.9

一般化非決定性有限オートマトンとは，以下で定義される「５つ組」$(Q, \Sigma, \delta, q_0, F)$ である．

Q	:	状態集合
Σ	:	入力アルファベット
δ	:	$(Q \setminus \{q_{\mathrm{accept}}\}) \times (Q \setminus \{q_{\mathrm{start}}\})$ から R への遷移関数
q_{start}	:	開始状態（$q_{\mathrm{start}} \in Q$）
q_{accept}	:	受理状態（$q_{\mathrm{accept}} \in Q$）

ただし，R は Σ 上のすべての正規表現の集合である．

まず，DFA M を一般化非決定性有限オートマトン G に変換する．

問 3.16. M から G への変換を示しなさい．

主張 3.1. G は L を認識する．

証明. 変換の手順より明らか． ■

次に，G から L を表す正規表現を求める手順を図 3.5 に示す．

主張 3.2. 図 3.5 のアルゴリズムにおいて，ステップ 1 の実行後の GNFA を G_{reg} とする．このとき，$L(G) = L(G_{\mathrm{reg}})$ である．

証明. G の状態の個数 k についての帰納法により示す．$k = 2$ のとき，主張が成り立つのは明らかである．状態の個数が $k - 1$ 以下の任意の GNFA について主張が成り立つとする．状態の個数が k の任意の GNFA を G とする．アルゴリズムのステップ 1 の一回の繰り返しにより変換された GNFA を G' とする．ステップ 1-(a) で選ばれた状態を q とする．帰納段階を示すためには，$L(G) = L(G')$ を示せばよい．

$(\subseteq)$ $w \in L(G)$ を任意とする．（つまり，w は G で受理される文字列.）$q_{i_1}, q_{i_2}, \ldots, q_{i_n}$ を，G が w を受理するときの状態の列とする．（$q_{i_1} = q_{\mathrm{start}}$.）これに対して，任意の $j \in [n-1]$ について，$w_j \in \delta(q_{i_j}, q_{i_{j+1}})$ とする．（$w_j \in \Sigma^*$.）このとき，w が G' で受理されることを示す．$q \notin \{q_{i_1}, q_{i_2}, \ldots, q_{i_n}\}$ のとき，G' が w を受理するのは明らかである．

問 3.17. この理由を説明しなさい．

入力：GNFA G

1. G が三つ以上の状態からなる GNFA である限り以下を実行する．
   (a) $q \in Q \setminus \{q_{\mathrm{start}}, q_{\mathrm{accept}}\}$ を任意の状態とする．
   (b) 以下のように $G' = (Q', \Sigma, \delta', q_{\mathrm{start}}, q_{\mathrm{accept}})$ を構成する．
      i. $Q' = Q \setminus \{q\}$.
      ii. 任意の $q_i \in Q \setminus \{q, q_{\mathrm{accept}}\}$, $q_j \in Q \setminus \{q, q_{\mathrm{start}}\}$ について，
      $$\delta'(q_i, q_j) \stackrel{\mathrm{def}}{=} R_1 \circ R^* \circ R_2 \cup R_3.$$
      ただし，
      $$\begin{array}{rcl} R & = & \delta(q, q) \\ R_1 & = & \delta(q_i, q) \\ R_2 & = & \delta(q, q_j) \\ R_3 & = & \delta(q_i, q_j) \end{array}$$
2. 変換された G の q_{start} から q_{accept} へのラベルを出力する．

図 3.5: 変換手順

$q \in \{q_{i_1}, q_{i_2}, \ldots, q_{i_n}\}$ のとき，$q = q_{i_j}$ とする．このとき，$w_{j-1} \in \delta(q_{i_{j-1}}, q_{i_j})$, $w_j \in \delta(q_{i_j}, q_{i_{j+1}})$ である．アルゴリズムのステップ 1-(b)-(ii) の δ' の構成より，$w_{j-1}w_j \in \delta'(q_{i_{j-1}}, q_{i_{j+1}})$ である．これより，w が G' により受理される．

($\supseteq$) $w \in L(G')$ を任意とする．（つまり，w は G' で受理される文字列．）$q_{i_1}, q_{i_2}, \ldots, q_{i_n}$ を，G' が w を受理するときの状態の列とする．（$q_{i_1} = q_{\mathrm{start}}$．）これに対して，任意の $j \in [n-1]$ について，$w_j \in \delta(q_{i_j}, q_{i_{j+1}})$ とする．（$w_j \in \Sigma^*$．）このとき，w が G で受理されることを示す．任意の $j \in [n-1]$ について，

以上より，$L(G) = L(G')$ となる．G' の状態の個数は $k-1$ であることから，帰納仮定より，$L(G') = L(G_{\mathrm{reg}})$．よって，$L(G) = L(G') = L(G_{\mathrm{reg}})$. ■

この主張より補題が示される． ■

3.5　非正規言語

定理 3.6 (ポンプの補題)**.** L を任意の正規言語とする．このとき，ある自然数 p が存在して，長さ p 以上の任意の $a \in L$ について，次のことが成り立つ．以下を満たす a の

分割 $a = xyz$ が存在する．

$$\begin{array}{ll} 1. & \forall i \in \mathbb{N} \cup \{0\}[xy^i z \in L] \\ 2. & |y| > 0 \\ 3. & |xy| \leq p \end{array}$$

証明． L が正規言語であることから，L を認識する有限オートマトン $M = (Q, \Sigma, \delta, q_0, F)$ が存在する．このとき，定理の自然数 p の値を $p = |Q|$ とする．長さが p 以上の $a \in L$ を任意に固定する*1．$|a| = n \geq p$ とする．$a \in L$ より，$r_0, r_1, \ldots, r_n \in Q$ が存在して，

$$\begin{array}{ll} 1. & r_0 = q_0 \\ 2. & \forall i \in [n][\delta(r_{i-1}, a_i) = r_i] \\ 3. & r_n \in F \end{array}$$

主張 3.3. ある $i, j \in [p] \cup \{0\}$（$i < j$）が存在して $r_i = r_j$．

証明． $r_0, r_1, \ldots, r_p$ を考える．$|Q| = p$ より，鳩ノ巣原理から，$r_i = r_j$ となる $i, j \in [p]$ が存在する． ■

上の主張を満たす最小の i，及び $r_i = r_j$ となる最小の j（$i < j$）を考える．このとき，a の分割 $a = xyz$ を以下のように定義する．

$$\begin{array}{lcl} x & \stackrel{\text{def}}{=} & a_1 \cdots a_i, \\ y & \stackrel{\text{def}}{=} & a_{i+1} \cdots a_j, \\ z & \stackrel{\text{def}}{=} & a_{j+1} \cdots a_p. \end{array}$$

このとき，以下の図より，任意の i について $xy^i z \in L$ であることは明らかである． ■

注 3.5. ポンプの補題の条件は，正規言語であるための必要条件を提示している．（十分条件ではない．）

例 3.14 (ポンプの補題)**.** 言語 $L = \{w \in \{0, 1\}^* : w$ は偶数個の 1 と偶数個の 0 からなる $\}$ は正規言語である．よって，ある自然数 p が存在して，長さ p 以上の任意の $w \in L$ について，補題の三つの条件を満たす w の分割 $w = xyz$ が存在する．例えば，$w = (01)^p \in L$ の場合，分割 $w = xyz$ は以下となる．

$$\begin{array}{lcl} x & = & \epsilon \\ y & = & (01)^2 \\ z & = & (01)^{p-2} \end{array}$$

*1 そのような文字列がなければ，定理が成り立つことは明らかである．

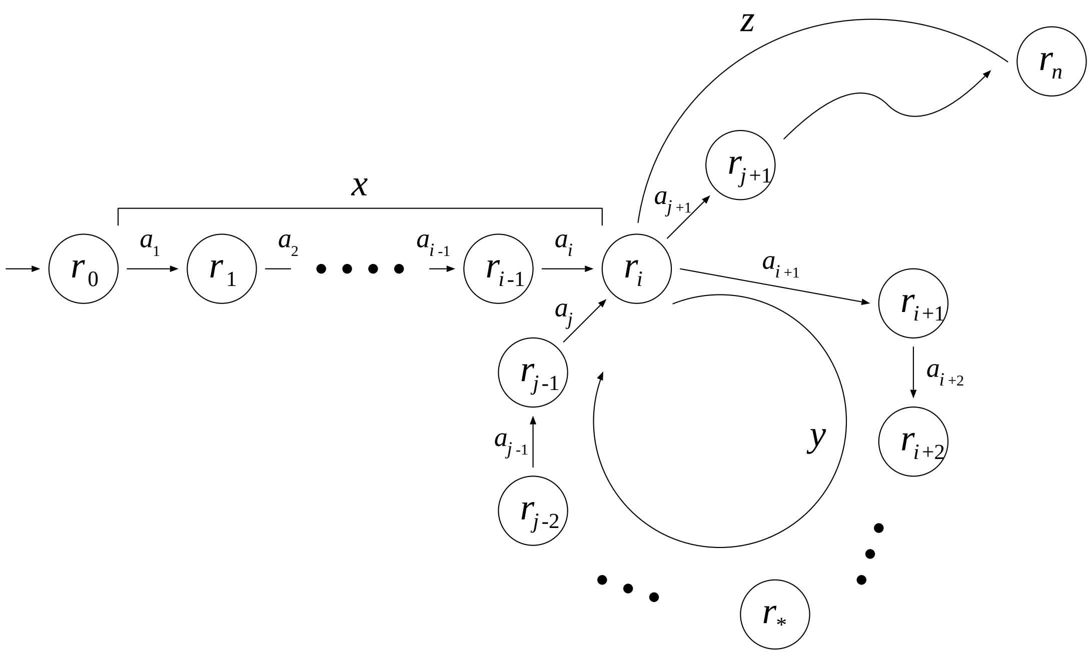


図 3.6: $xy^iz \in L$

このとき，補題の三つの条件は満たされる．また，他に，例えば，$w = 0^{2p}1^{2p} \in L$ の場合，分割 $w = xyz$ は以下となる．

$$\begin{array}{rcl} x &=& \epsilon \\ y &=& 0^2 \\ z &=& 0^{2p-2}1^{2p} \end{array}$$

このとき，補題の三つの条件は満たされる．

命題 3.7. $L = \{0^n1^n : n \in \mathbb{N}\}$ は非正規言語である．

証明. 背理法により示す．つまり，L が正規言語であるとする．このとき，ポンプの補題における自然数 p が存在する．長さが p 以上の $w \in L$ として，$w = 0^p1^p$ とする．以下，ポンプの補題にある三つの条件を満たす分割 $w = xyz = 0^p1^p$ が存在しないことを示す．三つ目の条件（$|xy| \leq p$）を考慮した場合，$xy = 0^i$, $z = 0^j1^p$ となる．（$i + j = p$.）このとき，二つ目の条件（$|y| > 0$）より $xyyz \notin L$ となる．一つ目の条件に反することから矛盾が導かれる．よって，L が非正規言語であることがいえる．

問 3.18. $xyyz \notin L$ である理由を説明しなさい．

■

問 3.19. 以下の言語が非正規言語であることを示しなさい．ただし，w^R は文字列 w を（左右）反転した文字列とする．

1. $L = \{ww : w \in \{0,1\}^*\}$.
2. $L = \{ww^R : w \in \{0,1\}^*\}$.
3. $L = \{w \in \{0,1\}^* : w = w^R\}$.

以上のように，正規言語ではないクラスが存在する．次章以降では，正規言語を真に含む言語のクラスを考える．

章末問題

以下の問いに答えなさい.

1. 以下の言語を受理するオートマトンを DFA または NFA（または，ϵ-NFA）で構成しなさい.
   （a）$L = \{w \in \{0,1\}^* : w$ は 111 を含まない $\}$.
   （b）$L = \{w \in \{0,1\}^* : \sum_{i=1}^{|w|} w_i \equiv_2 0\}$.
   （c）$L = \{w \in \{0,1,2\}^* : \sum_{i=1}^{|w|} w_i \equiv_3 0\}$.
   （d）正規表現 $\{0\} \circ \{0,1\}^* \circ \{1\}$ に対応する言語.
   （e）正規表現 $\{0\} \circ \{0,1\}^* \cup \{0,1\}^* \circ \{1\}$ に対応する言語.
2. 以下が非正規言語であることを示しなさい.（ポンプの補題を用いる.）
   （a）$L = \{0^i 1^j : i > j\}$.
   （b）$L = \{w \in \{0,1\}^* : w$ は同じ個数の 0 と 1 からなる $\}$.
   （c）$L = \{1^{n^2} : n \in \mathbb{N} \cup \{0\}\}$.
3. L_1, L_2, L を正規言語とする．このとき，以下が正規言語であることを説明しなさい.
   （a）$L_1 \cup L_2$.
   （b）$L_1 \cap L_2$.
   （c）$\overline{L}$.

第 4 章

文脈自由言語

文脈自由言語と呼ばれる，正規言語を真に含む言語のクラスを考え，文脈自由言語を生成する規則を考える．

4.1　文脈自由文法

定義 4.1

文脈自由文法とは，以下の「4 つ組」(V, Σ, R, S) である．

$$
\begin{array}{lcl}
V & : & \text{変数の集合} \\
\Sigma & : & \text{終端記号の集合} \\
R & : & \text{規則の集合} \\
S & : & \text{開始記号（}S \in V\text{）}
\end{array}
$$

集合 V, Σ, R はいずれも有限集合である．規則とは，$x \in V$, $y \in (V \cup \Sigma)^*$ に対して，$x \to y$ という表記のことである．（生成規則ともいう．）

例 4.1 (文脈自由文法)**.** 文脈自由文法 $G_1 = (V, \Sigma, R, S)$ を次のように定義する．$V = \{S\}$，$\Sigma = \{0, 1\}$ として，規則の集合 R を以下のように定義する．

$$
\begin{array}{lcl}
S & \to & 0S1, \\
S & \to & \epsilon.
\end{array}
$$

注 4.1. 上の例にある二つの規則は，以降，まとめて以下のように記述される．

$$S \quad \to \quad 0S1 \mid \epsilon.$$

例 4.2 (文脈自由文法)**.** 文脈自由文法 $G_2 = (V, \Sigma, R, S)$ を次のように定義する．$V =$

$\{S, A, B\}$，$\Sigma = \{0, 1\}$ として，規則の集合 R を以下のように定義する．

$$
\begin{array}{lcl}
S & \to & 0S1 \mid 0A \mid 1B, \\
A & \to & 0A \mid \epsilon, \\
B & \to & 1B \mid \epsilon.
\end{array}
$$

定義 4.2

$G = (V, \Sigma, R, S)$ を文脈自由文法とする．$u, v \in (V \cup \Sigma)^*$ が次の条件を満たすとき，u から v を**導出する**といい，$u \Rightarrow_G v$ と表記する．（G を省略して $u \Rightarrow v$ と表記することもある．）$x, y, z \in (V \cup \Sigma)^*$, $A \in V$ に対して，

$$
\begin{array}{ll}
1. & u = xAy, \\
2. & v = xzy, \\
3. & (A \to z) \in R.
\end{array}
$$

w を Σ 上の文字列とする．（つまり，$w \in \Sigma^*$．）G が w を**導出する**とは，$u_0, u_1, \ldots, u_n \in (V \cup \Sigma)^*$ が存在して，以下を満たすことである．

$$S = u_0 \Rightarrow u_1 \Rightarrow \cdots \Rightarrow u_{n-1} \Rightarrow u_n = w.$$

これを，$S \stackrel{*}{\Rightarrow}_G w$ と表記する．（G を省略して $S \stackrel{*}{\Rightarrow} w$ と表記することもある．）

例 4.3 (導出)**.** 例 4.1 の文脈自由文法 G_1 に対して，

$$
\begin{array}{lcl}
S & \Rightarrow_{G_1} & 0S1 \\
0S1 & \Rightarrow_{G_1} & 00S11 \\
00S11 & \Rightarrow_{G_1} & 000S111 \\
 & \vdots &
\end{array}
$$

これより，例えば，000111 が導出される．（$S \stackrel{*}{\Rightarrow}_{G_1} 000111$．）この導出列は以下である．

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 000111.$$

例 4.4 (導出)**.** 例 4.2 の文脈自由文法 G_2 に対して，

$$
\begin{array}{lcl}
00S11 & \Rightarrow_{G_2} & 000S111 \\
00S11 & \Rightarrow_{G_2} & 000A11 \\
00S11 & \Rightarrow_{G_2} & 001B11 \\
 & \vdots &
\end{array}
$$

これより, 例えば, 00011 や 00011111 が導出される．（$S \stackrel{*}{\Rightarrow}_{G_2} 00011$, $S \stackrel{*}{\Rightarrow}_{G_2} 00011111$．）これらの導出列はそれぞれ以下である．

$$
\begin{array}{lcl}
00011 & : & S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000A11 \Rightarrow 00011 \\
00011111 & : & S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000S111 \Rightarrow 0001B111 \Rightarrow 00011B111 \Rightarrow 00011111
\end{array}
$$

問 4.1. 例 4.1，例 4.2 の文脈自由文法 G_1, G_2 それぞれに対して，文字列 00111 は導出されるか．導出される場合は導出列を求めなさい．また，000011 はどうか．

定義 4.3

$G = (V, \Sigma, R, S)$ を文脈自由文法とする．G に対して，$L(G) \stackrel{\text{def}}{=} \{w \in \Sigma^* : S \stackrel{*}{\Rightarrow} w\}$ とする．このとき，G が $L(G)$ を**生成する**という．

Σ をアルファベット, $L \subseteq \Sigma^*$ を言語とする．ある文脈自由文法 $G = (V, \Sigma, R, S)$ が存在して $L = L(G)$ であるとき，L を**文脈自由言語**と呼ぶ．文脈自由言語のクラス（文脈自由言語の全集合）を $\mathcal{L}_{\mathrm{CFG}}$ と表記する．

例 4.5 (文脈自由言語)**.** 例 4.1 の文脈自由文法 G_1 に対して，

$$L(G_1) \ = \ \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}.$$

例 4.6 (文脈自由言語)**.** 例 4.2 の文脈自由文法 G_2 に対して，

$$L(G_2) \ = \ \{0^i 1^j : i, j \in \mathbb{N} \cup \{0\}, i \neq j\}.$$

命題 4.1. $\mathcal{L}_{\mathrm{REG}} \subsetneq \mathcal{L}_{\mathrm{CFG}}$.

証明. $\mathcal{L}_{\mathrm{REG}}$ が有限オートマトンにより認識できる言語のクラス $\mathcal{L}_{\mathrm{DFA}}$ に同等であること，また，$\mathcal{L}_{\mathrm{CFG}}$ がプッシュダウンオートマトンにより認識できる言語のクラス $\mathcal{L}_{\mathrm{PDA}}$ に同等であること（次章，定理 5.4），更に，$\mathcal{L}_{\mathrm{DFA}} \subsetneq \mathcal{L}_{\mathrm{PDA}}$（次章，命題 5.1），より示される．（実際，$L(G_1)$ や $L(G_2)$ が非正規言語であることから $\mathcal{L}_{\mathrm{REG}} \neq \mathcal{L}_{\mathrm{CFG}}$.） ■

例 4.7 (正規言語を生成する文脈自由文法)**.** $\Sigma = \{0, 1\}$ をアルファベットとする．正規言語 $L = \{w \in \Sigma^* : w$ は 0 で始まるか 1 で終わる $\}$ を生成する文脈自由文法 $G = (V, \Sigma, R, S)$ は次のようである．$V = \{S, A\}$ として，R は以下である．

$$\begin{array}{rcl} S & \to & 0A \mid A1, \\ A & \to & 0A \mid 1A \mid \epsilon. \end{array}$$

問 4.2. $\Sigma = \{0, 1\}$ をアルファベットとする．以下の正規言語を生成する文脈自由文法を求めなさい．

1. $L_1 = L(\Sigma^* \circ \{010\} \circ \Sigma^*)$

2. $L_2 = L(\Sigma^* \circ \{010\})$
3. $L_3 = L((\Sigma \circ \Sigma)^*)$

例 4.8 (文脈自由言語)**.** 文脈自由文法 $G = (V, \Sigma, R, S)$ を次のように定義する．$V = \{S\}$，$\Sigma = \{(,), 文\}$ として，規則の集合 R を以下のように定義する．

$$S \quad \to \quad SS \mid (S) \mid 文$$

この G が生成する言語は，かっこが入れ子になった文字列の集合である．

問 4.3. 上の例の文脈自由文法 G について，以下の文字列の導出列を求めなさい．

1. ((文) 文)
2. (文)((文)(文))

例 4.9 (文脈自由言語)**.** 文脈自由文法 $G = (V, \Sigma, R, < \mathrm{exp} >)$ を次のように定義する．$V = \{< \mathrm{exp} >, < \mathrm{term} >, < \mathrm{fact} >\}$，$\Sigma = \{+, -, *, /, (,)\} \cup \mathbb{R}$ として，規則の集合 R を以下のように定義する．

$$\begin{array}{lcl}
< \mathrm{exp} > & \to & < \mathrm{term} > + < \mathrm{exp} > \ \mid \ < \mathrm{term} > - < \mathrm{exp} > \ \mid \ < \mathrm{term} >, \\
< \mathrm{term} > & \to & < \mathrm{fact} > * < \mathrm{term} > \ \mid \ < \mathrm{fact} > / < \mathrm{term} > \ \mid \ < \mathrm{fact} >, \\
< \mathrm{fact} > & \to & (< \mathrm{exp} >) \mid a.
\end{array}$$

この G が生成する言語は，a を実数とした算術式の集合である．（次節の導出木の例を参照．）例えば，算術式 $1 - (2 - 3)$ の導出は，

$$\begin{aligned}
< \mathrm{exp} > &\Rightarrow < \mathrm{term} > - < \mathrm{exp} > \\
&\Rightarrow < \mathrm{fact} > - < \mathrm{term} > \\
&\Rightarrow 1- < \mathrm{fact} > \\
&\Rightarrow 1 - (< \mathrm{exp} >) \\
&\Rightarrow 1 - (< \mathrm{term} > - < \mathrm{exp} >) \\
&\Rightarrow 1 - (< \mathrm{fact} > - < \mathrm{term} >) \\
&\Rightarrow 1 - (2- < \mathrm{fact} >) \\
&\Rightarrow 1 - (2 - 3)
\end{aligned}$$

また，算術式 $(1 - (2 - 3)) * (-1) - 2 * 3/5$ の導出は，

$$\begin{aligned}
< \mathrm{exp} > &\Rightarrow < \mathrm{term} > - < \mathrm{exp} > \\
&\Rightarrow < \mathrm{fact} > * < \mathrm{term} > - < \mathrm{fact} > * < \mathrm{term} >
\end{aligned}$$

$$
\begin{aligned}
&\Rightarrow (< \text{exp} >)* < \text{fact} > -2* < \text{term} > \\
&\Rightarrow (1-(2-3))*(< \text{exp} >)-2* < \text{fact} > / < \text{term} > \\
&\Rightarrow (1-(2-3))*(< \text{term} >)-2*3/ < \text{fact} > \\
&\Rightarrow (1-(2-3))*(< \text{fact} >)-2*3/5 \\
&\Rightarrow (1-(2-3))*(-1)-2*3/5
\end{aligned}
$$

問 4.4. 以下の算術式の導出列を求めなさい.

1. $1-2-3$
2. $(1-2)*(2-3)/(3-1)$
3. $1+1/(1+1/(1+1/2))$

問 4.5. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の文脈自由文法 $G_i = (V_i, \Sigma, R_i, S)$ が生成する言語をそれぞれ求めなさい.

1. $V_1 = \{S, A\}$, $R_1 = \{S \to A1A, A \to A0|\epsilon\}$.
2. $V_2 = \{S\}$, $R_2 = \{S \to 0S0|1S1|\epsilon\}$.
3. $V_3 = \{S\}$, $R_3 = \{S \to 0S0|1S1|0|1|\epsilon\}$.

命題 4.2. アルファベットを $\Sigma = \{0, 1\}$ とする．$\{ww^R : w \in \{0, 1\}^*\}$, $\{w \in \Sigma^* : w = w^R\}$ は（非正規言語である）文脈自由言語である．一方，$\{ww : w \in \{0, 1\}^*\}$ は文脈自由言語でない.

証明. 上の問より，$\{ww^R : w \in \{0, 1\}^*\}$, $\{w \in \Sigma^* : w = w^R\}$ は文脈自由言語である．$\{ww : w \in \{0, 1\}^*\}$ は文脈自由言語でないことは，次章で示される（ポンプの補題より導かれる）命題 5.8 を参照. ■

問 4.6. $\Sigma = \{0, 1\}$ とする．以下の文脈自由言語を生成する文脈自由文法をそれぞれ求めなさい.（一つ目と二つ目は正規言語でもある．それ以外は非正規言語である.）

1. $L_1 = \{w \in \Sigma^* : w$ は 1 をちょうど二つ含む $\}$.
2. $L_2 = \{w \in \Sigma^* : w$ は 00 を含まない $\}$.
3. $L_3 = \{w \in \Sigma^* : w$ は奇数長で中央の文字が 1 である $\}$.

4. $L_4 = \{0^i 1^j : i > j\}$.
5. $L_5 = \{w \in \{0,1\}^* : w$ は同じ個数の 0 と 1 からなる $\}$.

以上の例や問でみてきた文脈自由文法や文脈自由言語は，それらが対応していることが（比較的）容易に分かるものであった．一方，以下の命題が示すように，ある言語が文脈自由言語であること，または，その生成規則が与えられたとしても，それらが対応するものであるかは明らかでないものがある．

命題 4.3. アルファベットを $\Sigma = \{0,1\}$ とする．言語 $L = \Sigma^* \setminus \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}$ は文脈自由言語である．

証明. 言語 L を生成する文脈自由文法 $G = (V, \Sigma, R, S)$ の規則 R が，以下であることを示す．（$V = \{S, A, B, X, Y\}$.）

$$\begin{array}{lcl} S & \to & XA \mid B \\ A & \to & XXA \mid \epsilon \\ X & \to & 1 \mid 0 \\ B & \to & 0B1 \mid Y \\ Y & \to & 0A0 \mid 1A0 \mid 1A1 \end{array}$$

つまり，$L(G) = L$ を示す．

（$\subseteq$）$w \in L(G)$ を任意の文字列とする．（$w \in L$ を示せばよい．）まず，$S \Rightarrow XA \Rightarrow \cdots \Rightarrow w$ の場合を考える．ここで，

$$A \Rightarrow XXA \Rightarrow XXXXA \Rightarrow \cdots$$

より，A からは偶数長の（任意の）文字列が導出される．よって，この場合，（S からは）奇数長の（任意の）文字列が導出されるので，$w \in L$ となる．次に，$S \Rightarrow B \Rightarrow \cdots \Rightarrow w$ の場合を考える．このとき，任意の導出において，ある整数 $i \in \mathbb{N} \cup \{0\}$ に対して，

$$S \Rightarrow B \Rightarrow 0B1 \Rightarrow 00B11 \Rightarrow \cdots \Rightarrow 0^i B 1^i \Rightarrow 0^i Y 1^i$$

となる．更に，以下の三つのいずれかが導出される．

$$\begin{array}{llcl} 1. & 0^i Y 1^i & \Rightarrow & 0^i 0A01^i \\ 2. & 0^i Y 1^i & \Rightarrow & 0^i 1A01^i \\ 3. & 0^i Y 1^i & \Rightarrow & 0^i 1A11^i \end{array}$$

前述のよう，A からは偶数長の（任意の）文字列が導出されるので，$w \in L$ となる．

(⊇)　$w \in L$ を任意の文字列とする．（$w \in L(G)$ を示せばよい．）$|w|$ が奇数の場合，(⊆) の証明より，w が $S \Rightarrow XA$ の規則により導出されることは明らかである．$|w|$ が偶数の場合，$w \in L$，つまり，$w \notin \{0^n1^n : n \in \mathbb{N} \cup \{0\}\}$ より，w はある整数 $i \in \mathbb{N} \cup \{0\}$，偶数長のある文字列 $x \in \Sigma^*$ について，以下のいずれかに等しい．

1. 0^i0x01^i
2. 0^i1x01^i
3. 0^i1x11^i

この場合，(⊆) の証明より，w が $S \Rightarrow B$ の規則により導出されることは明らかである．
■

4.2　導出木

定義 4.4

$G = (V, \Sigma, R, S)$ を文脈自由文法とする．根付き木 T が G の**導出木**であるとは，以下を満たすことである．

1. T の葉でない頂点には変数がラベル付けされる．
2. T の葉には変数か終端記号がラベル付けされる．
3. T の根には S がラベル付けされる．
4. T の葉でない任意の頂点 v に対して次のことが満たされる．$v_1, \ldots, v_k$ を v の子とする．v のラベルを A として，$v_1, \ldots, v_k$ のラベルをそれぞれ $A_1, \ldots, A_k$ とする．このとき，$A \to A_1 \ldots A_k$ は生成規則である．

注 4.2. 導出木は，構文解析木，構文木，解析木，などとも呼ばれる．

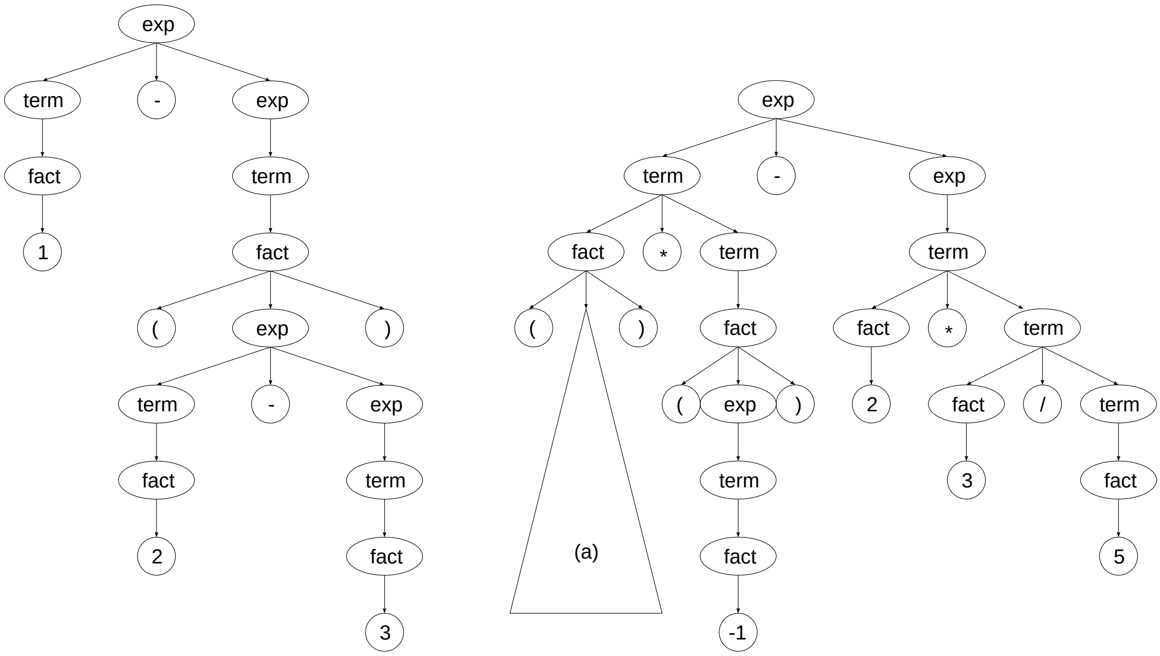
定義 4.5

$G = (V, \Sigma, R, S)$ を文脈自由文法とする．$w \in (V \cup \Sigma)^*$ とする．このとき，根付き木 T が w を導出する**導出木**であるとは，以下を満たすことである．

1. T が G の導出木である．
2. T を（左優先の）深さ優先探索をした場合，葉のラベルを探索順に並べると w となる．

例 4.10. 例 4.9 であげた算術式を考える．終端記号 a を任意の実数とした場合，(a) $1-(2-3)$，(b) $(1-(2-3))*(-1)-2*3/5$，を導出する導出木はそれぞれ以下の図のようになる．（右図の (b) において，三角形 (a) は，左図の (a) の導出木全体がそのまま

埋め込まれる.)



(a) $1-(2-3)$　(b) $(1-(2-3))*(-1)-2*3/5$

図 4.1: 算術式を導出する導出木

問 4.7. 以下の算術式を導出する導出木を示しなさい.

1. $1-2-3$
2. $(1-2)*(2-3)/(3-1)$
3. $1+1/(1+1/(1+1/2))$

4.3　チョムスキー標準形

定義 4.6

$G=(V,\Sigma,R,S)$ を文脈自由文法とする. R の任意の規則が以下の二つの形をしているとき, G をチョムスキー標準形という.

$$\begin{array}{ccc} A & \to & BC \\ A & \to & a \end{array}$$

ここで, A は変数, B,C は開始記号でない変数, a は終端記号とする.

定理 4.4 (チョムスキー標準形)**.** 任意の文脈自由文法は，チョムスキー標準形に変換できる．

L を Σ 上の任意の文脈自由言語とする．この章では，L を生成する規則，つまり，文脈自由文法を考えた．では，（正規言語と同様）Σ 上の任意の文字列 $x \in \Sigma^*$ が与えられたとき，文字列 x が言語 L に属するかどうか，つまり，$x \in L$ かどうかを判定するにはどうしたらよいだろうか？

章末問題

以下の問いに答えなさい．

1. 以下の文脈自由文法 $G = (V, \Sigma, R, S)$ が生成する文脈自由言語を示しなさい．
   $V = \{S, A, B\}$, $\Sigma = \{0, 1\}$,

$$\begin{array}{lcl} S & \to & AB \mid BA \\ A & \to & 0A0 \mid 0A1 \mid 1A0 \mid 1A1 \mid 0 \\ B & \to & 0B0 \mid 0B1 \mid 1B0 \mid 1B1 \mid 1 \end{array}$$

2. 以下の文脈自由言語の文脈自由文法を示しなさい．
   （a）$L_1 = \{0^i 1^j : i > 2j, i, j \in \mathbb{N} \cup \{0\}\}$.
   （b）$L_2 = \{0^i 1^i 2^j : i, j \in \mathbb{N} \cup \{0\}\}$.
   （c）$L_3 = \{0^i 1^j 2^i : i, j \in \mathbb{N} \cup \{0\}\}$.

第 5 章

プッシュダウンオートマトン

プッシュダウンオートマトンは有限オートマトンを拡張したものである．有限オートマトンと同様，プッシュダウンオートマトンにも決定性と非決定性がある．しかし，有限オートマトンとは異なり，プッシュダウンオートマトンでは決定性と非決定性とで言語の認識能力に差がある．（つまり，決定性では認識できない非決定性で認識できる言語が存在する[*1]．）ここでは，非決定性のプッシュダウンオートマトンのみを扱い，特に断らない限り，プッシュダウンオートマトンといえば非決定性であるものとする．

5.1　非決定性プッシュダウンオートマトン

定義 5.1

非決定性プッシュダウンオートマトン（**PDA**）とは，以下で定義される「6 つ組」$(Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, F)$ である．

Q	:	状態集合
Σ_ϵ	:	入力アルファベット
Γ_ϵ	:	スタックアルファベット
δ	:	$Q \times \Sigma_\epsilon \times \Gamma_\epsilon$ から $2^{Q \times \Gamma_\epsilon}$ への遷移関数
q_0	:	開始状態（$q_0 \in Q$）
F	:	受理状態の集合（$F \subseteq Q$）

集合 $Q, \Sigma_\epsilon, \Gamma_\epsilon, F$ はいずれも有限集合である．また，Γ_ϵ は，スタックの底を示す記号 \$ を含んでいるものとする．

直感的には，PDA は，「スタック」を備えた NFA である．

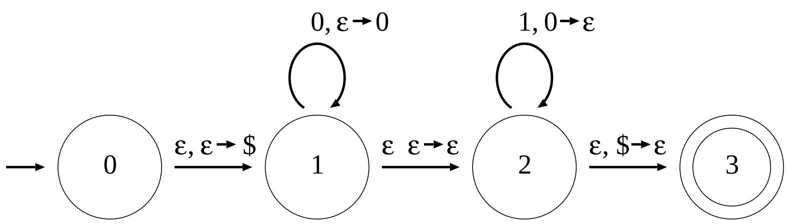
---

*1　$\{ww^R : w \in \{0,1\}^*\}$ がその例である．

例 5.1 (非決定性プッシュダウンオートマトン)**.** 非決定性プッシュダウンオートマトン $N_1 = (Q, \Sigma_\epsilon, \Gamma_\epsilon,\ \delta, q_0, \{q_3\})$ を次のように定義する．$Q = \{q_0, q_1, q_2, q_3\}$, $\Sigma = \{0, 1\}, \Gamma = \{0, \$\}$ として，遷移関数 $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon}$ を以下のように定義する．（空欄は ϕ を意味する．）

Σ_ϵ	0			1			ϵ		
Γ_ϵ	0	\$	ϵ	0	\$	ϵ	0	\$	ϵ
q_0									$(q_1, \$)$
q_1			$(q_1, 0)$			(q_2, ϵ)			
q_2				(q_2, ϵ)				(q_3, ϵ)	
q_3									

これを図示すると以下のようになる．（遷移先が ϕ の場合，その遷移は図中で省略される．）



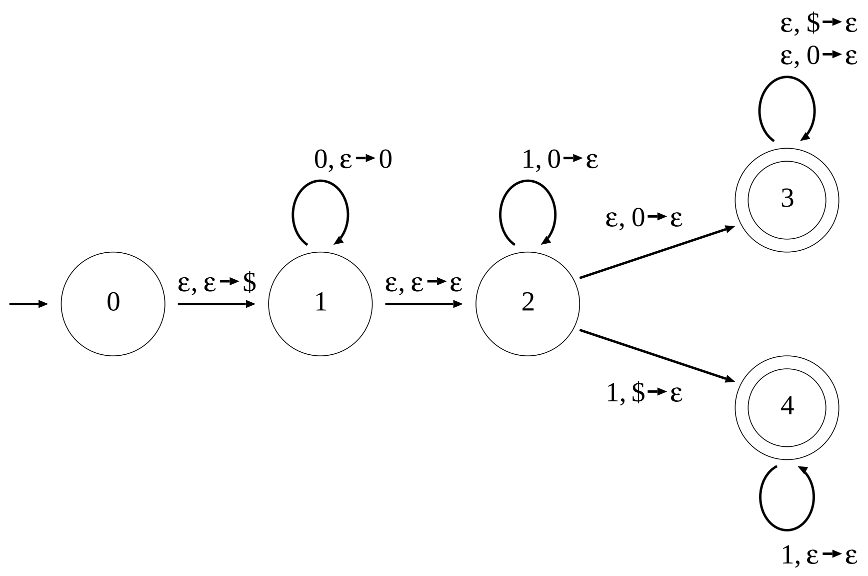
注 5.1. 遷移関数を表した図では，遷移関数 δ のある遷移，$\delta(q_i, y, a) = (q_j, b)$（$y \in \Sigma_\epsilon$, $a, b \in \Gamma_\epsilon$）は，$y, a \to b$ と表される．

例 5.2 (非決定性プッシュダウンオートマトン)**.** 非決定性プッシュダウンオートマトン $N_2 = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, \{q_3, q_4\})$ を次のように定義する．$Q = \{q_0, q_1, \ldots, q_4\}$, $\Sigma = \{0, 1\}, \Gamma = \{0, \$\}$ として，遷移関数 $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon}$ を以下の図で示されたように定義する．

定義 5.2

$N = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, F)$ をプッシュダウンオートマトンとする．$w = w_1 w_2 \ldots w_n$ を Σ 上の長さ n の文字列とする．（つまり，$w \in \Sigma^n$．）N が w を**受理する**とは，次を満たすことである．$w = y_1 \cdots y_m$（$y_i \in \Sigma_\epsilon$）と表せられて，状態の列 $r_0, r_1, \ldots, r_m \in Q$ と文字列 $s_0, s_1, \ldots, s_m \in \Gamma^*$ が存在して，

1. $r_0 = q_0$, $s_0 = \epsilon$, $s_1 = \$$.



2. $\forall i \in [m], \exists a, b \in \Gamma_\epsilon, \exists t \in \Gamma^* \left[\begin{array}{ll} 1. & (r_i, b) \in \delta(r_{i-1}, y_i, a) \\ 2. & s_{i-1} = at \\ 3. & s_i = bt \end{array} \right].$

3. $r_m \in F$.

N に対して，$L(N) = \{w \in \Sigma^* : N$ が w を受理する $\}$ とする．このとき，N が言語 $L(N)$ を**認識する**という．

非決定性プッシュダウンオートマトンが認識する言語のクラスを $\mathcal{L}_{\mathrm{PDA}}$ と表記する．（つまり，$\mathcal{L}_{\mathrm{PDA}} = \{L :$ ある PDA が L を認識する $\}$.）

注 5.2. 遷移先が $\emptyset$ である場合，（読み込まれていない入力文字列がまだあったとしても）その遷移の時点で受理されないことを意味する．（NFA の場合と同様.）

注 5.3. 遷移関数 δ のある遷移，$\delta(q_i, y, a) = (q_j, b)$（$y \in \Sigma_\epsilon$, $a, b \in \Gamma_\epsilon$）において，$a \to b$ は，「a を pop して b を push する」という「スタック操作」を意味する[*2]．特に，

$$\begin{array}{lcl} a = \epsilon & : & b \text{ を push} \\ b = \epsilon & : & a \text{ を pop} \end{array}$$

例 5.3 (非決定性プッシュダウンオートマトン)**.** 例 5.1 の非決定性有限オートマトンを N_1 とする．このとき，

$$\begin{array}{lcl} \epsilon,\ 01,\ 0011,\ 000111 & \in & L(N_1) \\ 0,\ 1,\ 000,\ 111,\ 0101,\ 1001,\ 00111 & \notin & L(N_1) \end{array}$$

[*2] Γ_ϵ がスタックアルファベットといわれる所以である．

つまり，N_1 が認識する言語 $L_1 = L(N_1)$ は，

$$L_1 \;=\; \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}.$$

例 5.4 (非決定性プッシュダウンオートマトン)**.** 例 5.2 の非決定性有限オートマトンを N_2 とする．このとき，

$$\begin{array}{lcl} 0,\ 1,\ 000,\ 111,\ 00011,\ 00111 & \in & L(N_2) \\ \epsilon,\ 01,\ 0011,\ 000111,\ 0101,\ 1001 & \notin & L(N_2) \end{array}$$

つまり，N_2 が認識する言語 $L_2 = L(N_2)$ は，

$$L_2 \;=\; \{0^i 1^j : i, j \in \mathbb{N} \cup \{0\}, i \neq j\}.$$

命題 5.1.

$$\mathcal{L}_{\mathrm{DFA}} \;\subsetneq\; \mathcal{L}_{\mathrm{PDA}}.$$

証明. $\mathcal{L}_{\mathrm{DFA}} \subseteq \mathcal{L}_{\mathrm{PDA}}$ であることは，DFA と PDA の定義より明らか.（DFA は PDA に「スタックを用いない」という制限を加えたオートマトンであるから.）$L = \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}$ とすれば，$L \in \mathcal{L}_{\mathrm{PDA}}$.（例 5.1, 5.3 より.）一方，$L \notin \mathcal{L}_{\mathrm{DFA}}$.（$L$ は非正規言語.）よって，$\mathcal{L}_{\mathrm{DFA}} \neq \mathcal{L}_{\mathrm{PDA}}$. ■

問 5.1. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の非決定性プッシュダウンオートマトンが認識する言語をそれぞれ求めなさい.（PDA を図示すると分かりやすくなる.）

1. $N_1 = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, \{q_3\})$ として，遷移関数を以下のように定義する.

Σ_ϵ	0				1				ϵ			
Γ_ϵ	0	1	\$	ϵ	0	1	\$	ϵ	0	1	\$	ϵ
q_0												$(q_1, \$)$
q_1				$(q_1, 0)$				$(q_1, 1)$				(q_2, ϵ)
q_2	(q_2, ϵ)					(q_2, ϵ)					(q_3, ϵ)	
q_3												

2. $N_2 = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, \{q_3\})$ として，遷移関数を以下のように定義する.

Σ_ϵ	0				1			
Γ_ϵ	0	1	\$	ϵ	0	1	\$	ϵ
q_0								
q_1				$(q_1, 0), (q_2, \epsilon)$				$(q_1, 1), (q_2, \epsilon)$
q_2	(q_2, ϵ)					(q_2, ϵ)		
q_3								

Σ_ϵ	ϵ			
Γ_ϵ	0	1	\$	ϵ
q_0				$(q_1, \$)$
q_1				(q_2, ϵ)
q_2			(q_3, ϵ)	
q_3				

問 5.2. アルファベットを $\Sigma = \{0, 1\}$ とする．以下の言語を認識するプッシュダウンオートマトンをそれぞれ求めなさい．

1. $L_1 = \{w \in \Sigma^* : w$ は 1 をちょうど二つ含む $\}$.
2. $L_2 = \{w \in \Sigma^* : w$ は奇数長で中央の文字が 1 である $\}$.
3. $L_3 = \{0^i 1^j : i > j, i, j \in \mathbb{N} \cup \{0\}\}$.
4. $L_4 = \{0^i 1^j : i < j, i, j \in \mathbb{N} \cup \{0\}\}$.
5. $L_5 = \{w \in \{0, 1\}^* : w$ は同じ個数の 0 と 1 からなる $\}$.

命題 5.2. PDA の遷移関数 $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon}$ を次のように拡張した．$\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon^*}$．これは，スタックの操作に文字列のプッシュができることを意味する．（一文字ではなく．）このように PDA を定義しても一般性を失わない．

命題 5.3. PDA の遷移関数 $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon}$ を以下の二種類に制限した．

- $\delta(q, *, \epsilon) = \{(q', *) : q' \in Q'\}$.
- $\delta(q, *, a) = \{(q', \epsilon) : q' \in Q'\}$.

これは，スタックの操作がプッシュかポップしかできないことを意味する．（書き換えができない．）このように PDA を定義しても一般性を失わない．

5.2 文脈自由言語との等価性

定理 5.4.

$$\mathcal{L}_{\mathrm{CFG}} = \mathcal{L}_{\mathrm{PDA}}.$$

この定理を示すためには，$\mathcal{L}_{\mathrm{CFG}} \subseteq \mathcal{L}_{\mathrm{PDA}}$ かつ $\mathcal{L}_{\mathrm{CFG}} \supseteq \mathcal{L}_{\mathrm{PDA}}$ を示せばよい．

補題 5.5 ($\mathcal{L}_{\mathrm{CFG}} \subseteq \mathcal{L}_{\mathrm{PDA}}$)**.** L を任意の文脈自由言語とする．L を認識するプッシュダウンオートマトンが存在する．

証明.（概要）L が文脈自由言語であることから，ある文脈自由文法 $G = (V, \Sigma, R, S)$ が存在して $L = L(G)$ である．以下，G を模倣するプッシュダウンオートマトン $P = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, F)$ の構成を示す．$Q = \{q_0, q_{loop}, q_1\}$, $\Gamma = V \cup \Sigma$, $F = \{q_1\}$ とする．命題 5.2 より，遷移関数は $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to 2^{Q \times \Gamma_\epsilon^*}$ として以下で定義される．

- $\delta(q_0, \epsilon, \epsilon) = \{(q_{loop}, S)\}$.
- それぞれの規則 $A \to B$ （$A \in V \subseteq \Gamma$, $B \in \Gamma_\epsilon^*$）について，$\delta(q_{loop}, \epsilon, A) = \{(q_{loop}, B)\}$.
- それぞれの終端記号 a について，$\delta(q_{loop}, a, a) = \{(q_{loop}, \epsilon)\}$.
- $\delta(q_{loop}, \epsilon, \$) = \{(q_1, \epsilon)\}$.

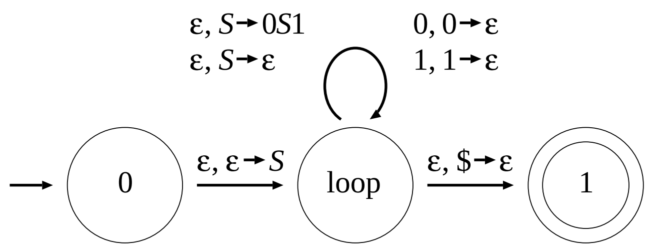
主張 5.1. $L = L(P)$.

証明.（概要）遷移関数の定義より，導出木を左優先で展開していることが分かる．（また，G が導出しない文字列は P で受理しない．）そのことから，P が G を正確に模倣していることがいえる．よって，$L = L(P)$. ■

この主張より，ある PDA P が存在して，P が L を認識することがいえる． ■

例 5.5. 例 4.1 で示された CFG を PDA P にすると，以下の図で示されたようになる．このとき，$L = L(P) = \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}$.

問 5.3. 例 4.2 で示された CFG を PDA に変換しなさい．



補題 5.6 ($\mathcal{L}_{\mathrm{CFG}} \supseteq \mathcal{L}_{\mathrm{PDA}}$)**.** $P = (Q, \Sigma_\epsilon, \Gamma_\epsilon, \delta, q_0, F)$ を任意のプッシュダウンオートマトン，L を P が認識する言語とする．L は文脈自由言語である．

証明. 命題 5.3 より，P の遷移関数は，スタックの操作がプッシュかポップのみであるとする．更に，一般性を失うことなく，P が以下であるとする．

- $|F| = 1$.（受理状態は唯一である.）
- 受理状態ではスタックは空である.

以下，P を模倣する文脈自由文法 $G = (V, \Sigma, R, S)$ の構成を示す．$V = \{A_{pq} : p, q \in Q\} \cup \{S\}$ とする．R は以下で定義される．

- $S \to A_{q_0 q_{accept}}$.
- ある $c \in \Gamma$ が存在して，$(s, c) \in \delta(p, a, \epsilon)$ かつ $(q, \epsilon) \in \delta(t, b, c)$ を満たす，任意の $p, q, s, t \in Q$, 任意の $a, b \in \Sigma$ に対して，$A_{pq} \to a A_{st} b$.
- 任意の $p, q, r \in Q$ について $A_{pq} \to A_{pr} A_{rq}$.
- 任意の $p \in Q$ について $A_{pp} \to \epsilon$.

主張 5.2. 任意の $x \in \Sigma^*$，任意の $p, q \in Q$ について，次が成り立つ．入力 x により，スタックがからの状態 p からスタックが空の状態 q へ遷移するならば，A_{pq} から x が導出される．

証明. $|x|$ の長さについての帰納法により示す．$x \in \Sigma^*$，$p, q \in Q$ を任意に固定する．x を（p を開始状態とした）P に入力したとき，p から q までの状態遷移の一つを $p = r_0, r_1, \ldots, r_k = q$ とする．$r_1, \ldots, r_{k-1}$ の状態の中で，スタックが空になった状態を $r_{i_1}, \ldots, r_{i_\ell}$ とする．また，それぞれの状態へ遷移するときに読み込んだ x を $x_{j_1}, \ldots, x_{j_\ell}$ とする．このとき，任意の $h \in [\ell - 1]$ について，入力 $x_{j_h} \ldots x_{j_{h+1}}$ により，状態 r_{i_h} から状態 $r_{i_{h+1}}$ へ遷移する．帰納仮定より，$A_{r_{i_h} r_{i_{h+1}}}$ から文字列 $x_{j_h} \ldots x_{j_{h+1}}$ が導出される．また，R の定義より，$A_{r_0 r_k}$ から $A_{r_0 r_{i_1}} \cdots A_{r_{i_\ell} r_k}$ が導出される．よって，A_{pq}（$= A_{r_0 r_k}$）から x が導出されることが示される． ■

主張 5.3. 任意の $p, q \in Q$，任意の $x \in \Sigma^*$ について，次が成り立つ．A_{pq} から x が導出されるならば，入力 x により，スタックが空の状態 p からスタックが空の状態 q へ遷移する．

証明. $|x|$ の長さについての帰納法により示す．$x \in \Sigma^*$，$p, q \in Q$ を任意に固定する．$|x| = k$ のときを考える．R の定義より，x の導出には以下の二つの場合がある．

- $A_{pq} \to aA_{st}b$
- $A_{pq} \to A_{pr}A_{rq}$

一つ目のとき，ある $c \in \Gamma$ が存在して，（$x_1 = a$, $x_k = b$ に対して）$(s, c) \in \delta(p, a, \epsilon)$ かつ $(q, \epsilon) \in \delta(t, b, c)$ である．これは，入力 x_1 により，スタックが空の状態 p からスタックが c だけの状態 s へ，また，入力 x_k により，スタックが c だけの状態 t からスタックが空の状態 q へ遷移することを意味する．仮定より，A_{st} から $x_2 \ldots x_{k-1}$ が導出されるので，帰納仮定より，入力 $x_2 \ldots x_{k-1}$ により，スタックが空の状態 s からスタックが空の状態 t へ遷移する．これらより，入力 x により，スタックが空の状態 p からスタックが空の状態 q へ遷移することが示される．

二つ目のとき，ある状態 r が存在して，また，仮定より x のある分割 $x = yz$ が存在して，A_{pr} により y が，A_{rq} により z が導出される．帰納仮定より，入力 y により，スタックが空の状態 p からスタックが空の状態 r へ，また，入力 z により，スタックが空の状態 r からスタックが空の状態 q へ遷移する．これらより，入力 x により，スタックが空の状態 p からスタックが空の状態 q へ遷移することが示される．■

主張 5.4. $L = L(G)$.

証明. それぞれの主張について，$p = q_0$，$q = q_{accept}$ とすれば，一つ目の主張より $L \subseteq L(G)$ が，二つ目の主張より $L \supseteq L(G)$ が，それぞれ示される．よって，$L = L(G)$ が示される．■

この主張より，ある CFG G が存在して，G が L を生成することがいえる．■

例 5.6. $L = \{0^n 1^n : n \in \mathbb{N}\}$ を認識する PDA （例 5.1 の PDA は受理状態が二つ）を CFG にすると，次のようになる．$G = (V, \Sigma, R, S)$ として，R は以下のよう：

- $S \to A_{03}$
- $$\begin{array}{lcl} A_{03} & \to & A_{12} \\ A_{12} & \to & 0A_{12}1 \big| 0A_{11}1 \end{array}$$
- 任意の $p, q, r \in \{0, 1, 2, 3\}$ について $A_{pq} \to A_{pr}A_{rq}$.

- 任意の $p \in \{0,1,2,3\}$ について $A_{pp} \to \epsilon$.

ただし，不必要な変数（A_{00}, A_{01}, A_{10} など）及びそれを含んだ規則（$A_{13} \to A_{12}A_{23}$, $A_{00} \to \epsilon$ など）は削除しておく必要がある.

問 5.4. 例 5.1 で示された PDA を変更して，受理状態が一つになる PDA を求めなさい．更に，その PDA を CFG に変換しなさい.

5.3　非文脈自由言語

定理 5.7 (ポンプの補題)**.** L を任意の文脈自由言語とする．このとき，ある自然数 p が存在して，長さ p 以上の任意の $w \in L$ について，次のことが成り立つ．以下を満たす w の分割 $w = axyzb$ が存在する.

1. $\forall i \in \mathbb{N} \cup \{0\}[ax^iyz^ib \in L]$
2. $|xz| > 0$
3. $|xyz| \leq p$

証明. L が文脈自由言語であることから，ある文脈自由文法 $G = (V, \Sigma, R, S)$ が存在して $L = L(G)$．また，定理 4.4 より，G がチョムスキー標準形であると仮定してよい．$p = 2^{|V|+1}$ と定義する．長さ p 以上の $w \in L$ を任意に固定する．w を導出する導出木 T を任意に固定する.

主張 5.5. T は二分木である.

問 5.5. 上の主張を証明しなさい.

主張 5.6. T の深さは $|V| + 1$ 以上である.

問 5.6. 上の主張を証明しなさい.

w の導出木 T において根から最も長い経路を P とする．上の主張より，P をなす頂点の個数は（根を含めて）$|V| + 2$ 以上である.

主張 5.7. ある変数 $A \in V$ が存在して，P の頂点のラベルに A が 2 回以上出現する．

証明. P の頂点のうち，葉以外の頂点のラベルはすべて V の変数であり，その個数は $|V|+1$ 以上である．よって，鳩ノ巣原理より，ある変数 $A \in V$ が存在して，P の頂点のラベルに A が 2 回以上出現する． ■

P の頂点を葉から順に $v_{|V|+1}, v_{|V|}, \ldots, v_1, v_0$，それに対応する頂点のラベルを $a, A_{|V|}, \ldots, A_1, A_0$ とする．（このとき，v_0 が根であるとは限らない！）更に，2 回出現する変数を A として，$A_i = A_j = A$（$i < j$）する．v_i を根とする T の部分木を T_i，v_j を根とする T の部分木を T_j とする．これを図示すると以下のようになる．

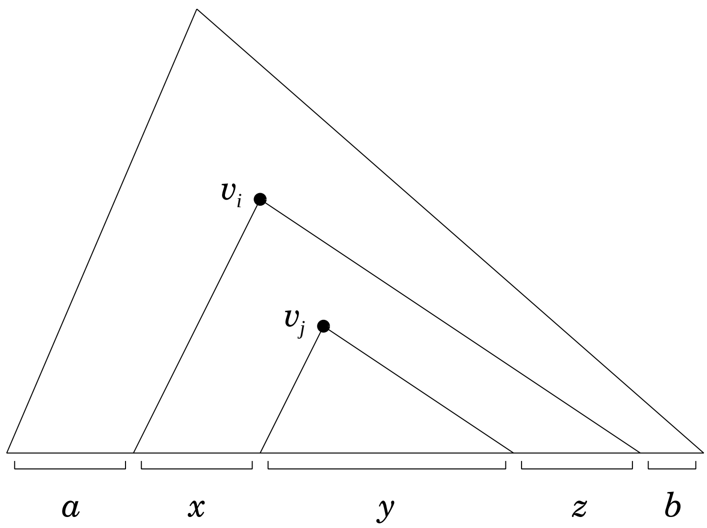


図 5.1: 導出木

主張 5.8. 部分木 T_i の深さは高々 $|V|+1$ である．

問 5.7. 上の主張を証明しなさい．

上の図のように，$w = axyzb$ と分割する．この分割が定理の三つの条件を満たすことを示す．まず，定理の条件 1 が満たされることを示す．v_i と v_j のラベルが同じであることから，v_i から文字列 y が導出されることを意味する．これは，以下の図で示されるよう，v_i から v_j までを短絡させることができることを意味する．つまり，ayb が導出されることになり，$ayb \in L$ であることが分かる．また，v_i と v_j のラベルが同じであることから，v_j から文字列 xyz が導出されることを意味する．これは，以下の図で示されるよう，v_j に v_i 以下の部分木 T_i を継ぎ足すことができることを意味する．つまり，ax^2yz^2b

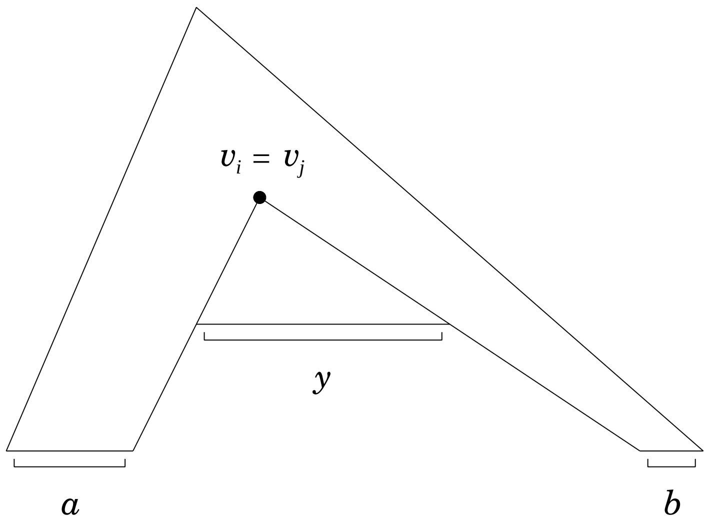


図 5.2: 導出木

が導出されることになり，$ax^2yz^2b \in L$ であることが分かる．（$ax^iyz^ib \in L$ も同様にして示される．）以上より，定理の条件 1 が満たされることが示される．

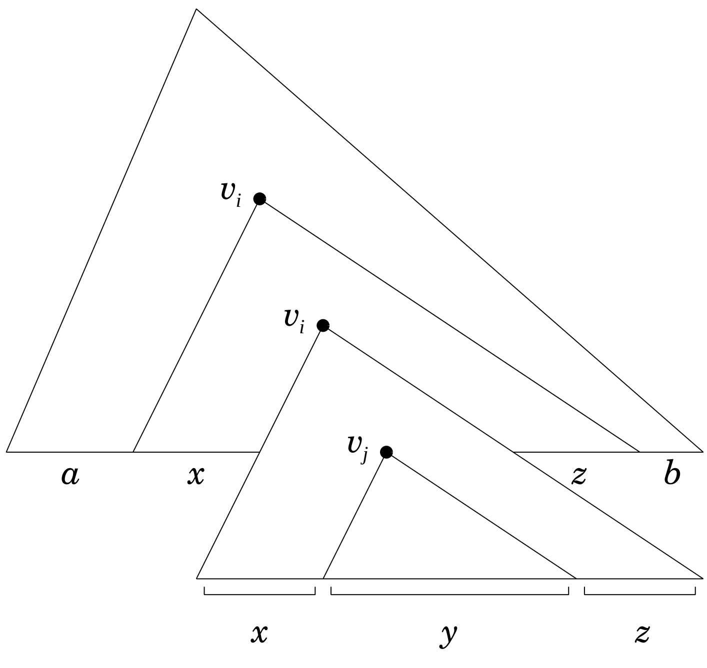


図 5.3: 導出木

次に，定理の条件 2 が満たされることを背理法により示す．$|xz| = 0$ とする．つまり，$x = z = \epsilon$ であるとする．このことは，$v_i = v_j$ であることを意味する．これは，$v_i \neq v_j$ であることに反する．よって，$|xz| > 0$ である．

最後に，定理の条件 3 が満たされることを示す．主張 5.8 より，T_i の深さは高々 $|V|+1$ である．これは，$|xyz| \leq 2^{|V|+1} = p$ を意味する．■

例 5.7 (ポンプの補題)**.** 言語 $L = \{ww^R : w \in \{0,1\}^*\}$ は文脈自由言語である．（命題 4.2 を参照.）よって，ある自然数 p が存在して，長さ p 以上の任意の $w \in L$ について，補題の三つの条件を満たす w の分割 $w = axyzb$ が存在する．例えば，$w = 0^p1^p1^p0^p \in L$ の場合，分割 $w = axyzb$ は以下となる．

$$\begin{array}{rcl} a & = & 0^p1^{\frac{p}{2}} \\ x & = & 1^{\frac{p}{2}} \\ y & = & \epsilon \\ z & = & 1^{\frac{p}{2}} \\ b & = & 1^{\frac{p}{2}}0^p \end{array}$$

このとき，補題の三つの条件は満たされる．

命題 5.8. $\{ww : w \in \Sigma^*\}$ は非文脈自由言語である．

証明. 背理法により示す．つまり，L が文脈自由言語であるとする．このとき，ポンプの補題における自然数 p が存在する．長さが p 以上の $w \in L$ として，$w = 0^p1^p0^p1^p$ とする．以下，ポンプの補題にある三つの条件を満たす分割 $w = axyzb = 0^p1^p0^p1^p$ が存在しないことを示す．三つ目の条件（$|xyz| \leq p$）を考慮した場合，文字列 xyz は以下のいずれかである．

$$xyz = \begin{cases} 0^i & i \leq p \\ 1^i & j \leq p \\ 0^i1^j & i+j \leq p \\ 1^i0^j & i+j \leq p \end{cases}$$

いずれの場合も，$axxyzzb \notin L$ となる．一つ目の条件に反することから矛盾が導かれる．よって，L が非文脈自由言語であることがいえる．

問 5.8. $axxyzzb \notin L$ である理由を説明しなさい．

■

問 5.9. $L = \{0^n1^n2^n : n \in \mathbb{N} \cup \{0\}\}$ が非文脈自由言語であることを示しなさい．

章末問題

以下の問いに答えなさい.

1. 文脈自由言語を認識するプッシュダウンオートマトンを求めなさい.（(e), (f) はチャレンジ問題.）
   (a) $L_1 = \{0^i 1^i 2^j, i, j \in \mathbb{N} \cup \{0\}\}$.
   (b) $L_2 = \{0^i 1^j 2^j : i, j \in \mathbb{N} \cup \{0\}\}$.
   (c) $L_3 = \{0^i 1^j 2^i : i, j \in \mathbb{N} \cup \{0\}\}$.
   (d) $L = \Sigma^* \setminus \{0^n 1^n : n \in \mathbb{N} \cup \{0\}\}$.
   (e)* 例 4.8 であげた文脈自由言語.
   (f)* 例 4.9 であげた文脈自由言語.
2. 以下が非文脈自由言語であることを示しなさい.
   (a) $L = \{0^i 1^j 2^k : 0 \leq i \leq j \leq k\}$.
   (b) $L = \{0^{n^2} : n \in \mathbb{N} \cup \{0\}\}$.
3. L_1, L_2, L を任意の文脈自由言語とする．このとき，以下は文脈自由言語であるか.理由も述べなさい.
   (a) $L_1 \cup L_2$.
   (b) $L_1 \cap L_2$.
   (c) $\overline{L}$.

問の略解

導入

1. Σ 上の文字列として abcabca，そうでないものとして abcdef.
2. Σ は Σ 上のある言語である.
3. Σ^k は Σ の文字を k 個並べた文字列全体の集合であることから，それら可能な文字列の個数は n^k となる.
4. $\Sigma \circ \Sigma^*$ は，長さが 1 以上の文字列の集合.（つまり，$\epsilon \notin \Sigma \circ \Sigma^*$．一方，$\epsilon \in \Sigma^*$.）
5. (a) $\epsilon \in \Sigma^0 \subseteq \Sigma^*$ より.
   (b) $\emptyset^0 = \{\epsilon\}, \forall k \in \mathbb{N}[\emptyset^k = \emptyset]$ より.

正規表現

1. (a) 任意の言語は空集合との和をとっても変わらないので.
   (b) ϵ は一つの文字列であるので.
2. (a) 任意の $w \in L$ について $w\epsilon = \epsilon w = w$ なので.
   (b) $xy \in L \circ \emptyset$ と $x \in L$ かつ $y \in \emptyset$ は同値なので.（$y \in \emptyset$ は成り立たない.）
3. $L^* = \{\epsilon, 00, 11, 0000, 0011, 1100, 1111, 000000, \ldots\}$．これは，$\Sigma = \{00, 11\}$ をアルファベットとすれば，$L^* = \Sigma^*$.
4. 言語のすべての要素が有限長であれば，それぞれの要素は Σ の要素（を集合としたもの）と $\circ$ で表記できる．また，言語の大きさは有限であるため，それらの和集合とした表記にすれば（その言語の）正規表現となる.
5. それぞれの正規表現を示せばよい．$\{0^n : n \in \mathbb{N}\}$ の正規表現は $\{0\} \circ \{0\}^*$ である．同様に，$\{1^n : n \in \mathbb{N}\}$ の正規表現は $\{1\} \circ \{1\}^*$ である．また，$\{0^i 1^j : i, j \in \mathbb{N}\}$ の正規表現は $\{0\} \circ \{0\}^* \circ \{1\} \circ \{1\}^*$ である.
6. (a) $\{w \in \Sigma^* : w$ は最初と最後が同じ文字列 $\}$.
   (b) $\{w \in \Sigma^* : w$ は偶数長の文字列 $\}$.

(c) $\{w \in \Sigma^* : w$ は偶数個の 0 を含む $\}$.

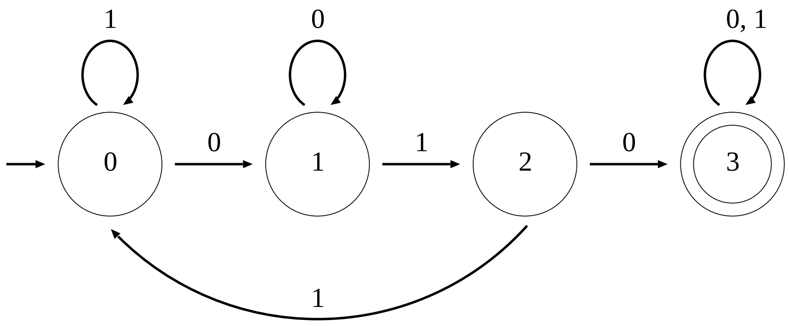
7. (a) $\{0\} \circ \Sigma^* \circ \{1\} \cup \{1\} \circ \Sigma^* \circ \{0\}$.
   (b) $\Sigma \circ (\Sigma \circ \Sigma)^*$.
   (c) $\{1\}^* \circ \{0\} \circ \{1\}^* \circ (\{0\} \circ \{1\}^* \circ \{0\} \circ \{1\}^*)^*$.
8. $\{1\}^*$ が $\{1^n : n \in \mathbb{N} \cup \{0\}\}$ であることから，$\{01\} \circ \{1^n : n \in \mathbb{N} \cup \{0\}\}$ は $\{011^n : n \in \mathbb{N} \cup \{0\}\}$ となる．
9. A の要素が何回か繰り返された後，空列である場合と，0 が一つ続く場合があるので．
10. 略．
11. 略．
12. $\{1\}^* \circ (\{01, 001\} \circ \{1\}^*)^* \circ \{\epsilon, 0, 00\}$

章末問題

1. (a) $\{1\}^* \circ \{0\}^*$
   (b) $\{0\}^* \circ \{1\}^*$
   (c) $\{0\}^* \circ (\{10\} \circ \{0\}^*)^* \circ \{\epsilon, 1\}$
   (d) $\{1\}^* \circ (\{01\} \circ \{1\}^*)^* \circ \{0\}^*$
   (e) 略．
2. (a) $\{\epsilon\}$
   (b) $\{\epsilon, 1\} \circ \{01\}^* \circ \{\epsilon, 0\}$
   (c) $\{\epsilon, 1, 11\} \circ (\{0, 00\} \circ \{1, 11\})^* \circ \{\epsilon, 0, 00\}$
3. (a) $\{w \in \Sigma^* : w$ は 0 と 1 が交互に現れる文字列 $\}$.
   (b) $\{w \in \Sigma^* : w$ は 00 を含まない $\}$.
   (c) $\{w \in \Sigma^* : w$ は 000 を含まない $\}$.

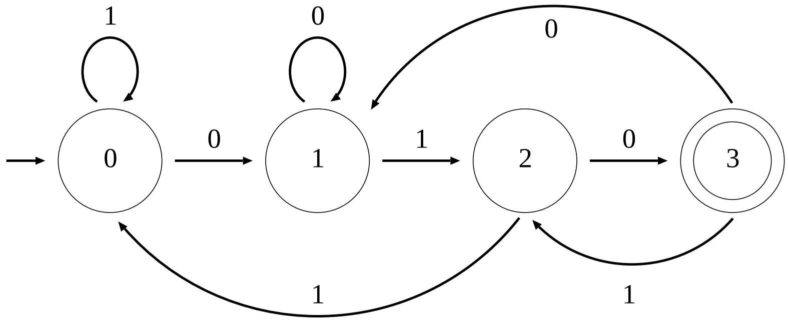
オートマトン

1. (a) $\{w \in \Sigma^* : w$ は 1 をちょうど一つ含む $\}$
   (b) $\{w \in \Sigma^* : w$ は偶数個の 1 を含む $\}$
   (c) $\{w \in \Sigma^* : w$ は 3 の倍数個の 1 を含む $\}$
2. (a) 010 を含む：



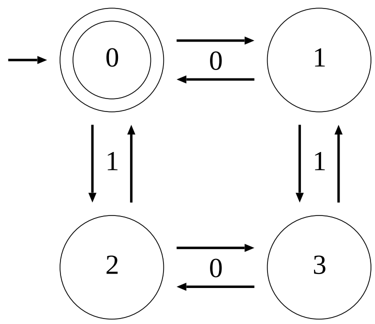
（b）略.

（c）010 で終わる：



（d）略.

（e）偶数個：

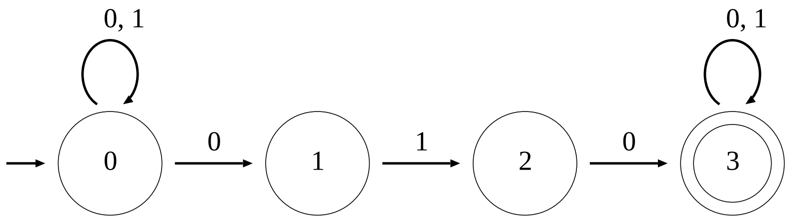


3. $\delta(q_0, 1) = \{q_0, q_1\}$ という遷移.

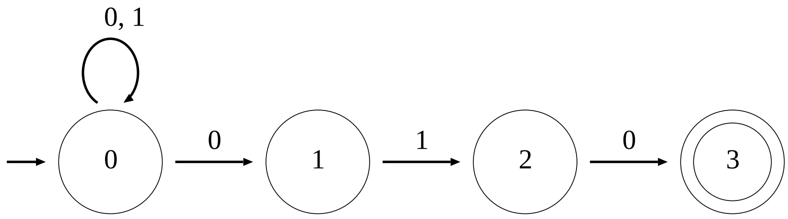
4. （a）$\{w \in \Sigma^* : w$ は 0 で始まり 1 で終わる $\}$

（b）$\{w \in \Sigma^* : w$ は 1 で始まり 0 で終わる $\}$

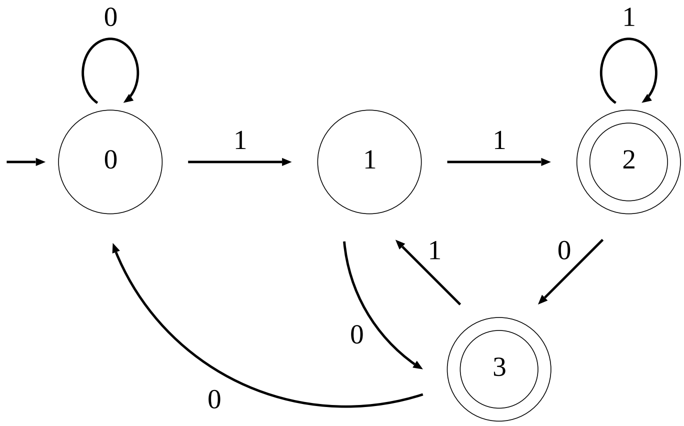
5. （a）010 を含む：



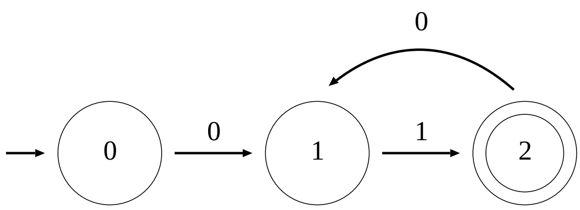
（b）010 で終わる：



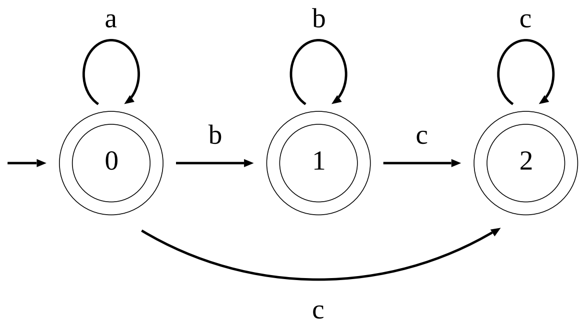
6. 可能である．以下がその一例．



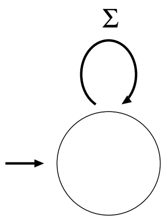
7. DFA で認識できれば，NFA で認識できるので．
8. 認識の定義より N が x を受理するのは，$\delta^*(q_0, x) \cap F \neq \emptyset$ であることである．一方，D が x を受理するのは，$\delta_D^*(q_D, x) \in F_D$ であることである．F_D の定義より，$\delta^*(q_0, x) = \delta_D^*(q_D, x)$ であれば，N が受理するときかつそのときに限り D が受理する．
9. $|x| = 0$ は $x = \epsilon$ を意味する．よって，$\delta_D^*(q_D, x) = \delta^*(q_0, x) = \{q_0\}$．
10. （a）略．
    （b）略．
11. 可能である．以下がその一例．



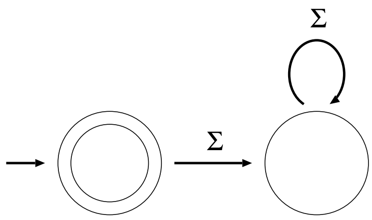
12. 可能である．以下がその一例．ただし，遷移はすべて決定的であるが，NFA の一つとして記述されている．（それゆえ，省略されている遷移がある．）



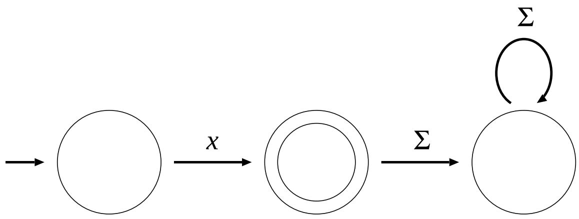
13. 上の二つを参照.
14. それぞれ以下のようになる.
    (a) $L=\emptyset$



    (b) $L=\epsilon$



    (c) $L=\{x\}$



15. 各 $i=0,1,2$ について, $N_i=(Q_i,\Sigma,\delta_i,p_i,F_i)$ とする. このとき, $N=(Q,\Sigma,\delta,q_0,F)$ の遷移関数 δ は, もともとの δ_i に以下の遷移が加わる.
    (a) $\delta(q_0,\epsilon)=\{q_1,q_2\}$.
    (b) 任意の $q\in F_1$ について $\delta(q,\epsilon)=\{p_2\}$.
    (c) $\delta(q_0,\epsilon)=\{p_0\}$, 任意の $q\in F_0$ について $\delta(q,\epsilon)=\{p_0\}$.
16. M の開始状態 q_0 と受理状態の集合 F, G の開始状態 q_{start} と受理状態 q_{accept} に対して, $\delta(q_{\mathrm{start}},q_0)=\{\epsilon\}$, 任意の $q\in F$ について $\delta(q,q_{\mathrm{accept}})=\{\epsilon\}$ とする. また, $\delta(q,a)=q'$ について $\delta(q,q')=\{a\}$ とする. それ以外の q,q' ($q\neq q_{\mathrm{accept}}$,

$q' \neq q_{\mathrm{start}}$）について $\delta(q, q') = \emptyset$ とする．

17. G' においても，w を入力したときの状態の列は G と同じになるので．
18. $xyyz = 0^k 1^p$ で $k > p$ となるので．
19. （a）$1^p 0^p 1^p 0^p \in L$ の任意の分割 $1^p 0^p 1^p 0^p = xyz$ を考える．$|xy| \le p$ より，$xy = 1^i$, $z = 1^j 0^p 1^p 0^p$ である．このとき，$xyyz \notin L$ となる．
    （b）$1^p 0^p 0^p 1^p \in L$ の任意の分割 $1^p 0^p 0^p 1^p = xyz$ を考える．$|xy| \le p$ より，$xy = 1^i$, $z = 1^j 0^p 0^p 1^p$ である．このとき，$xyyz \notin L$ となる．
    （c）$0^p 1 0^p \in L$ の任意の分割 $0^p 1 0^p = xyz$ を考える．$|xy| \le p$ より，$xy = 0^i$, $z = 0^j 1 0^p$ である．このとき，$xyyz \notin L$ となる．

章末問題

1. （a）言語 $L' = \{w \in \{0,1\}^* : w$ は 111 を含む $\}$ を認識する DFA の受理状態を非受理にまたその逆をすればよい．
    （b）言語 L の補足：たして 2 の倍数．ただし，$w_i \in \{0, 1\}$.
    （c）言語 L の補足：たして 3 の倍数．ただし，$w_i \in \{0, 1, 2\}$.
    （d）略．
    （e）略．
2. （a）$a = 0^{p+1} 1^p$ とする．
    （b）$a = 0^p 1^p$ とする．
    （c）$a = 1^{p^2}$ とする．
3. L_1, L_2, L を認識する決定性有限オートマトンをそれぞれ M_1, M_2, M とする．
    （a）集合の和は正規演算の一つであるので．（または，M_1 と M_2 を「並列に」接続すれば，$L_1 \cup L_2$ を認識する（ϵ-非決定性）有限オートマトンとなる．）
    （b）$L_1 \cap L_2 = \overline{\bar{L}_1 \cup \bar{L}_2}$ より，(c) から $\bar{L}_1, \bar{L}_2$ が，(a) から $\bar{L}_1 \cup \bar{L}_2$ が，更に，(c) から $\overline{\bar{L}_1 \cup \bar{L}_2}$ がそれぞれ正規言語となる．
    （c）M の状態集合を Q，受理状態の集合を F としたとき，受理状態の集合が $Q \setminus F$ である M は $\bar{L}$ を認識するので．

文脈自由言語

1. G_1 からは双方ともに導出されない．G_2 からは両方とも導出される．その導出列は，

$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 001B11 \Rightarrow 00111$$
$$S \Rightarrow 0S1 \Rightarrow 00S11 \Rightarrow 000A11 \Rightarrow 0000A11 \Rightarrow 000011$$

2. (a) $V_1 = \{S, A\}$, $R_1 = \{S \to A010A, A \to A0|A1|\epsilon\}$.

   (b) $V_2 = \{S, A\}$, $R_2 = \{S \to A010, A \to A0|A1|\epsilon\}$.

   (c) $V_3 = \{S, A\}$, $R_3 = \{S \to AAS|\epsilon, A \to 0|1\}$.

3. (a) $S \Rightarrow (S) \Rightarrow (SS) \Rightarrow ((S)\ 文) \Rightarrow ((文)\ 文)$

   (b) $S \Rightarrow SS \Rightarrow (S)(S) \Rightarrow (文)(SS) \Rightarrow (文)((S)(S)) \Rightarrow (文)((文)(文))$

4. (a)

$$
\begin{aligned}
< \mathrm{exp} > &\Rightarrow < \mathrm{term} > - < \mathrm{exp} > \\
&\Rightarrow < \mathrm{fact} > - < \mathrm{term} > - < \mathrm{exp} > \\
&\Rightarrow 1- < \mathrm{fact} > - < \mathrm{term} > \\
&\Rightarrow 1 - 2- < \mathrm{fact} > \\
&\Rightarrow 1 - 2 - 3
\end{aligned}
$$

   (b)

$$
\begin{aligned}
< \mathrm{exp} > &\Rightarrow < \mathrm{term} > \\
&\Rightarrow < \mathrm{fact} > * < \mathrm{term} > \\
&\Rightarrow (< \mathrm{exp} >)* < \mathrm{fact} > / < \mathrm{term} > \\
&\Rightarrow (< \mathrm{term} > - < \mathrm{exp} >) * (< \mathrm{exp} >)/ < \mathrm{fact} > \\
&\Rightarrow (< \mathrm{fact} > - < \mathrm{term} >) * (< \mathrm{term} > - < \mathrm{exp} >)/(< \mathrm{exp} >) \\
&\Rightarrow (1- < \mathrm{fact} >) * (< \mathrm{fact} > - < \mathrm{term} >)/(< \mathrm{term} > - < \mathrm{exp} >) \\
&\Rightarrow (1 - 2) * (2- < \mathrm{fact} >)/(< \mathrm{fact} > - < \mathrm{term} >) \\
&\Rightarrow (1 - 2) * (2 - 3)/(3- < \mathrm{fact} >) \\
&\Rightarrow (1 - 2) * (2 - 3)/(3 - 1)
\end{aligned}
$$

   (c)

$$
\begin{aligned}
< \mathrm{exp} > &\Rightarrow < \mathrm{term} > + < \mathrm{exp} > \\
&\Rightarrow < \mathrm{fact} > + < \mathrm{term} > \\
&\Rightarrow 1+ < \mathrm{fact} > / < \mathrm{term} > \\
&\Rightarrow 1 + 1/ < \mathrm{fact} > \\
&\Rightarrow 1 + 1/(< \mathrm{exp} >) \\
&\overset{*}{\Rightarrow} 1 + 1/(1 + 1/(< \mathrm{exp} >)) \quad (\because\ \text{上記までの導出を適用する}) \\
&\Rightarrow 1 + 1/(1 + 1/(< \mathrm{term} > + < \mathrm{exp} >)) \\
&\Rightarrow 1 + 1/(1 + 1/(< \mathrm{fact} > + < \mathrm{term} >)) \\
&\Rightarrow 1 + 1/(1 + 1/(1+ < \mathrm{fact} > / < \mathrm{term} >)) \\
&\Rightarrow 1 + 1/(1 + 1/(1 + 1/ < \mathrm{fact} >)) \\
&\Rightarrow 1 + 1/(1 + 1/(1 + 1/2))
\end{aligned}
$$

5. (a) $L_1 = \{w \in \Sigma^* : w$ は 1 をちょうど一つ含む $\}$

（b）$L_2 = \{ww^R : w \in \{0,1\}^*\}$

（c）$L_3 = \{w \in \{0,1\}^* : w = w^R\}$

6. （a）$V_1 = \{S, A\}$, $R_1 = \{S \to A1A1A, A \to A0|\epsilon\}$.

（b）$V_2 = \{S, A\}$, $R_2 = \{S \to 1S|0A|\epsilon, A \to 1S|\epsilon\}$.

（c）$V_3 = \{S, A\}$, $R_3 = \{S \to ASA|1, A \to 0|1\}$.

（d）$V_4 = \{S, A\}$, $R_4 = \{S \to 0S1|0A, A \to 0A|\epsilon\}$.

（e）$V_5 = \{S\}$, $R_5 = \{S \to 0S1S|1S0S|\epsilon\}$.

以下，$L_5 = L(G_5)$ の証明の概略を示す．$L_5 \subseteq L(G_5)$ かつ $L_5 \supseteq L(G_5)$ を示せばよい．後者は明らか．前者を示すためには，任意の $w \in L_5$ について $w \in L(G_5)$ を示せばよい．$w = w_1 w_2 \ldots w_n \in \{0,1\}^n$ とする．一般性を失うことなく $w_1 = 0$ とする．このとき，ある文字列 $x, y \in L_5$ に対して $w_2 \ldots w_n = x1y$ となる．更に，この事実を x, y に繰り返し適用すれば，w が G_5 から生成されることが分かる．

7. 略.

章末問題

1. $L(G) = \{xy \in \Sigma^* : |x| = |y| \wedge x \neq y\}$.
2. （a）$V_1 = \{S, A\}$, $R_1 = \{S \to 00S1|0A, A \to 0A|\epsilon\}$.

（b）$V_2 = \{S, A, B\}$, $R_2 = \{S \to AB, A \to 0A1|\epsilon, B \to 2B|\epsilon\}$.

（c）$V_3 = \{S, A\}$, $R_3 = \{S \to 0S2|A, A \to 1A|\epsilon\}$.

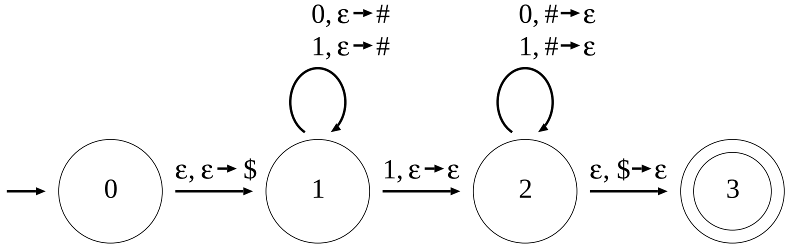
プッシュダウンオートマトン

1. （a）$L_1 = \{ww^R : w \in \Sigma^*\}$.

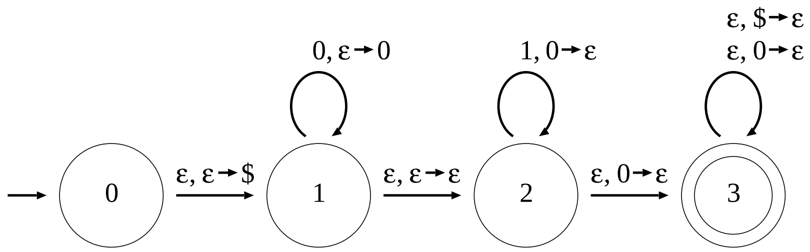
（b）$L_2 = \{w \in \Sigma^* : w = w^R\}$.

2. （a）略.

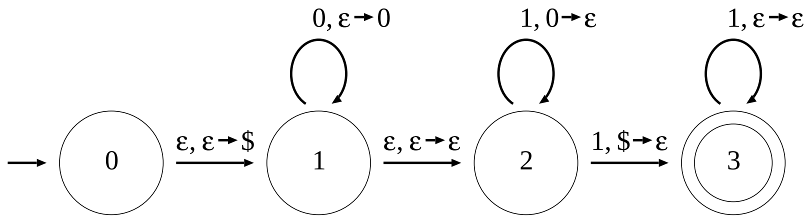
（b）



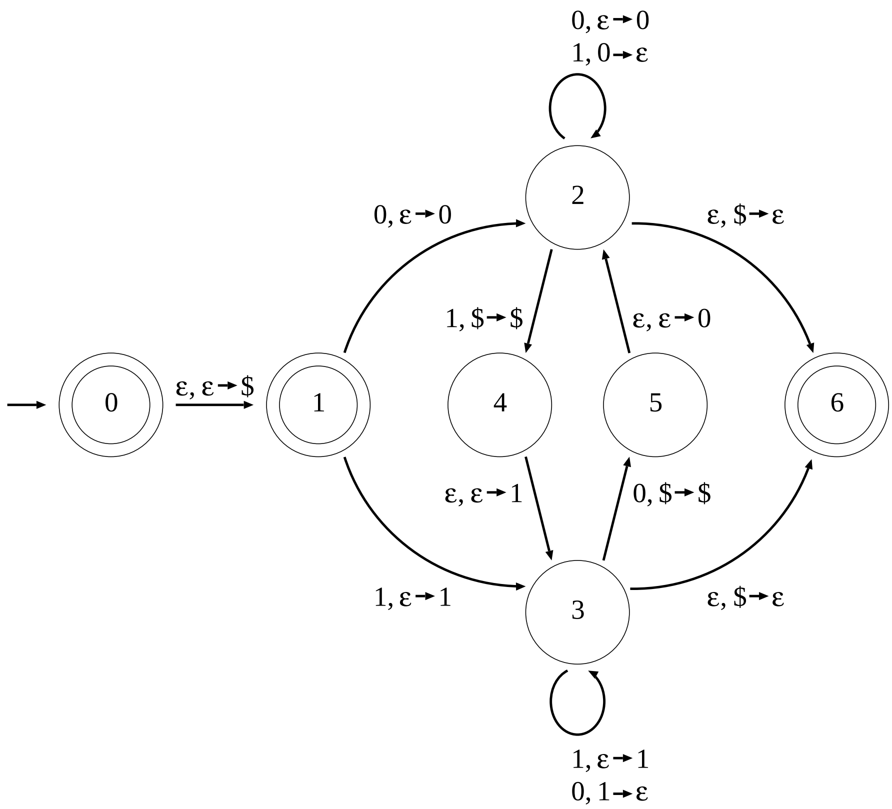
（c）



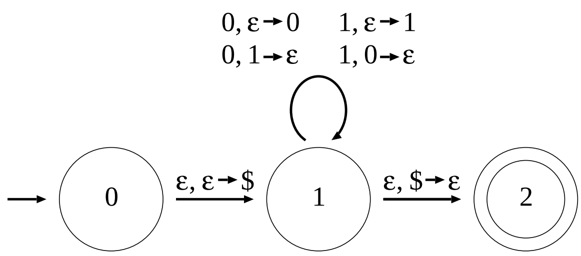
（d）



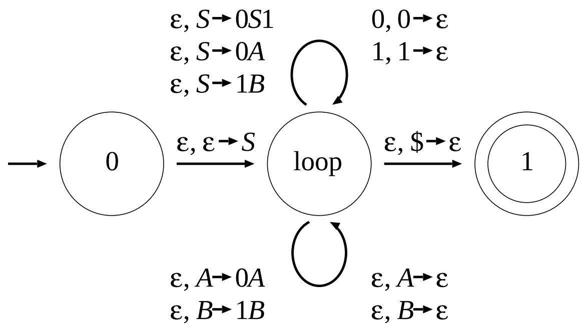
（e）



（別解）問 4.6 の 5 の解答を参照.



3.



4. まず，受理状態を一つにするために，例 5.1 の遷移関数を変更して $\delta(q_0, \epsilon, \epsilon) = \{(q_1, \$), (q_3, \epsilon)\}$ とする．例 5.6 との違いは，$A_{03} \to A_{12}|A_{00}|A_{33}$ である．（よって，A_{00}, A_{33} は必要な変数となる．）
5. 規則が $A \to BC$ か $A \to a$ の形式のみなので．
6. $|w| \geq p$ より，w の導出木の葉の個数は p 以上である．導出木の高さを ℓ とすれば，$2^\ell \geq p = 2^{|V|+1}$．これより，$\ell \geq |V| + 1$．
7. 経路 $v_i, v_{i+1}, \ldots, v_{|V|+1}$ の長さが T_i の深さとなるので．
8. $xyz = 0^i$（$i \leq p$）のとき，（$|xz| > 0$ より）ある $n > p$ に対して，$axxyzzb = 0^n 1^p 0^p 1^p \notin L$ となる．他の三つの場合も同様にして示される．
9. 長さが p 以上の $w \in L$ として，$w = 0^p 1^p 2^p$ を考える．

章末問題

1. （a）略．
   （b）略．
   （c）略．
2. （a）長さが p 以上の $w \in L$ として，$w = 0^p 1^p 2^p$ を考える．
   （b）長さが p 以上の $w \in L$ として，$w = 0^{p^2}$ を考える．$w = axyzb$ として，ポンプの補題より $|xyz| \leq p$ であるので，$|ax^2yz^2b| \leq p^2 + 2p < (p+1)^2$ より矛盾がいえる．
3. （a）文脈自由言語．L_1 の文脈自由言語の開始記号を S_1，L_2 の文脈自由言語の開始記号を S_2 とすれば，$S \to S_1|S_2$ とすればよい．
   （b）必ずしも文脈自由言語であるとはいえない．$L_1 = \{0^i 1^j 2^j : i, j \in \mathbb{N} \cup \{0\}\}$，$L_2 = \{0^j 1^j 2^i : i, j \in \mathbb{N} \cup \{0\}\}$ としたとき，（L_1, L_2 は文脈自由言語であ

る[*3]が）$L_1 \cap L_2 = \{0^n 1^n 2^n : n \in \mathbb{N} \cup \{0\}\}$ となるので．

（c）必ずしも文脈自由言語であるとはいえない．補集合について閉じているとする．$L_1 = \{0^i 1^j 2^j : i, j \in \mathbb{N} \cup \{0\}\}$, $L_2 = \{0^j 1^j 2^i : i, j \in \mathbb{N} \cup \{0\}\}$ とする．このとき，$\overline{L_1}, \overline{L_2}$ はともに文脈自由言語となり，$L = \overline{L_1} \cup \overline{L_2}$ も文脈自由言語となる．更に，$\overline{L}$ も文脈自由言語となる．しかし，（ド・モルガンの法則より）$\overline{L} = L_1 \cap L_2$ であり，これは，非文脈自由言語であることに反する．

別の例では，以下の二つは互いに補集合（$L_a = \overline{L_b}$）であり，L_a は非文脈自由言語（命題 5.8），L_b は文脈自由言語である[*4]．

- $L_a = \{ww : w \in \{0,1\}^*\}$.
- $L_b = \{w \in \{0,1\}^* : |w| \equiv_2 1\} \cup \{xy \in \{0,1\}^* : |x| = |y| \wedge x \neq y\}$.

[*3] 第 4 章の章末問題を参照.

[*4] 第 4 章の章末問題を参照.

参考図書

本テキストは，オートマトンと言語理論のごく一部（の初歩的なこと）しか扱っていない．オートマトンと言語理論について更に学びたい学生は，以下の教科書や，そこであげられている参考図書を参照するとよい．

1. 計算理論の基礎（1. オートマトンと言語），太田和夫他訳，共立出版，２００８年．（原書：Introduction to the Theory of Computation, Michael Sipser, Cengage Learning, 2005.）
2. 形式言語とオートマトン，守屋悦朗著，サイエンス社，２００１年．
3. オートマトン・言語理論・計算論（第２版），J. ホップクロフト，R. モトワニ，J. ウルマン著，野崎昭弘他訳，２００３年．

索引