有限オートマトン（07/03 配布）

(決定性) 有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ の形式的定義

- Q：有限集合 $\cdots$ 状態の集合
- Σ：有限集合 $\cdots$ 入力文字の集合（“alphabet”）
- $\delta : Q \times \Sigma \longrightarrow Q$：遷移関数
- $s \in Q$ $\cdots$ 初期状態
- $F \subset Q$ $\cdots$ 受理状態の集合

(決定性) 有限オートマトンによる語の受理

有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ が語 $w = a_1 a_2 \cdots a_n \in \Sigma^*$ を受理 (accept) する
$\Longleftrightarrow \exists r_0, r_1, \ldots, r_n \in Q$:

- $r_0 = s$
- $\delta(r_{i-1}, a_i) = r_i$ $(i = 1, \ldots, n)$
- $r_n \in F$

非決定性有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ の形式的定義

- Q：有限集合 $\cdots$ 状態の集合
- Σ：有限集合 $\cdots$ alphabet,　$\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$
- $\delta : Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$：遷移関数　$\cdots$ 可能な遷移先全体の集合
  ($\mathcal{P}(Q)$ は Q の冪集合)
- $s \in Q$ $\cdots$ 初期状態
- $F \subset Q$ $\cdots$ 受理状態の集合

非決定性有限オートマトンによる語の受理

非決定性有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ が語 $w \in \Sigma^*$ を受理する
$\Longleftrightarrow \exists a_1, a_2, \cdots, a_n \in \Sigma_\varepsilon, \exists r_0, r_1, \ldots, r_n \in Q$:

- $w = a_1 a_2 \cdots a_n$
- $r_0 = s$
- $r_i \in \delta(r_{i-1}, a_i)$ $(i = 1, \ldots, n)$
- $r_n \in F$

決定性有限オートマトンと非決定性有限オートマトンとの同等性

与えられた非決定性有限オートマトン $M = (Q, \Sigma, \delta, s, F)$ に対し、M が認識する言語 $L(M)$ を認識する決定性有限オートマトン $\widetilde{M} := (\widetilde{Q}, \Sigma, \widetilde{\delta}, \widetilde{s}, \widetilde{F})$ が次で構成できる：

まず、各状態 $q \in Q$ に対し、$E(q) \subset Q$ を次で定める：

$$E_0(q) := \{q\}, \quad E_{i+1}(q) := \bigcup_{r \in E_i(q)} \delta(r, \varepsilon), \quad E(q) := \bigcup_{i \geq 0} E_i(q)$$

($E(q)$ は状態 q から入力を何も読まずに遷移できる状態全体)

- $\widetilde{Q} := \mathcal{P}(Q)$：現在あり得る状態全体の集合
- $\widetilde{\delta} : \widetilde{Q} \times \Sigma \longrightarrow \widetilde{Q}$　：$\widetilde{q}$ の何処かから入力 x で遷移できる状態全体

$$(\widetilde{q}, x) \longmapsto \widetilde{\delta}(\widetilde{q}, x) := \bigcup_{q \in \widetilde{q}} \left(\bigcup_{r \in \delta(q, x)} E(r) \right)$$

- $\widetilde{s} := E(s)$
- $\widetilde{F} := \{\widetilde{q} \in \widetilde{Q} \mid \widetilde{q} \cap F \neq \emptyset\}$ ：あり得る状態のどれかが受理状態

正規言語に対する Pumping Lemma（注入補題・反復補題）

正規言語 A に対し、

$\exists n \in \boldsymbol{N} : \forall w \in A, |w| \geq n : \exists x, y, z \in \Sigma^*, w = xyz$ s.t.

(1) $y \neq \varepsilon$
(2) $|xy| \leq n$
(3) $\forall k \geq 0 : xy^k z \in A$

非決定性プッシュダウンオートマトン $M = (Q, \Sigma, \Gamma, \delta, s, F)$ の形式的定義

- Q：有限集合 $\cdots$ 状態の集合
- Σ：有限集合 $\cdots$ alphabet, $\Sigma_\varepsilon := \Sigma \cup \{\varepsilon\}$
- Γ：有限集合 $\cdots$ stack alphabet, $\Gamma_\varepsilon := \Gamma \cup \{\varepsilon\}$
- $\delta : Q \times \Sigma_\varepsilon \times \Gamma_\varepsilon \longrightarrow \mathcal{P}(Q \times \Gamma_\varepsilon)$：遷移関数（非決定的）　$\cdots$ 可能な遷移先全体の集合
- $s \in Q$ $\cdots$ 初期状態
- $F \subset Q$ $\cdots$ 受理状態の集合

生成文法における Chomsky の標準形

- 正規言語の生成規則は次の形に出来る
  - $\star$ $X \longrightarrow xY$ $(X, Y \in V, x \in \Sigma)$
  - $\star$ $X \longrightarrow x$ $(X \in V, x \in \Sigma_\varepsilon)$
- 文脈自由言語の生成規則は次の形に出来る
  - $\star$ $X \longrightarrow YZ$ $(X, Y, Z \in V)$
  - $\star$ $X \longrightarrow x$ $(X \in V, x \in \Sigma_\varepsilon)$

文脈自由言語に対する Pumping Lemma

文脈自由言語 A に対し、

$\exists n \in \boldsymbol{N} : \forall w \in A, |w| \geq n : \exists u, v, x, y, z \in \Sigma^* : w = uvxyz$ s.t.

(1) $vy \neq \varepsilon$
(2) $|vxy| \leq n$
(3) $\forall k \geq 0 : uv^k xy^k z \in A$

主なレポート課題の例（07/03 配布）

問 1. 言語 $L \subset \Sigma^*$ に対し、$S_L : \Sigma^* \longrightarrow \mathcal{P}(\Sigma^*)$ を $S_L(w) := \{v \in \Sigma^* | wv \in L\}$ で定める。このとき、L が有限オートマトンで認識可能 $\Longleftrightarrow$ $\mathrm{Im} S_L$ が有限集合。

問 2. 正規言語が非決定性有限オートマトンで認識可能であることを示すために、次の言語を認識する非決定性有限オートマトンを形式的に（状態遷移の概念図ではなく、5 つ組 $M = (Q, \Sigma, \delta, s, F)$ を与えることによって）構成せよ。

(1) $L(\emptyset) = \emptyset, L(\varepsilon) = \{\varepsilon\}, L(a) = \{a\}$ $(a \in \Sigma)$
(2) 非決定性有限オートマトン $M_i = (Q_i, \Sigma, \delta_i, s_i, F_i)$ が認識する言語 $L_i = L(M_i)$ に対し、$L_1 \cup L_2, L_1 L_2, L_1^*$

問 3. Σ を alphabet とする正規言語 $A \in \mathcal{P}(\Sigma^*)$ に対し、A に属さない語全体から成る言語 $\Sigma^* \setminus A$ も再び正規言語であることを示せ。（ヒント：正規表現の定義から示すのは煩わしい。A を認識する有限オートマトン M_A が存在するので、これを用いて $\Sigma^* \setminus A$ を認識する有限オートマトンを構成せよ。）

問 4. Σ を alphabet とする正規言語 $A, B \in \mathcal{P}(\Sigma^*)$ に対し、$A \cap B$ も再び正規言語であることを示せ。（ヒント：正規表現の定義から示すのは煩わしい。A, B を認識する有限オートマトン M_A, M_B を用いて $A \cap B$ を認識する有限オートマトンを構成するか、前問と de Morgan の法則を用いよ。）

問 5. $\Sigma = \{a, b\}$ を alphabet とする次の言語について、(a) 正規表現で表せ。(b) 生成規則で表せ。(c) 受理する非決定性有限オートマトンを構成せよ。(d) 受理する決定性有限オートマトンを構成せよ。(e) 上記の有限オートマトンを模倣するプログラムを作成せよ（言語は何でも良い）。

(1) a で始まり b で終わる
(2) a, b の片方しか現われない
(3) a が 3 つ以上続かない
(4) a が偶数個
(5) その他、<u>適当に非自明で興味深い</u>例を考えよ。

問 6. $\Sigma = \{a, b\}$ を alphabet とする次の言語について、(a) 正規言語でないことを示せ（正規言語に対する Pumping Lemma（注入補題・反復補題）または部屋割り論法を用いよ）。(b) 生成規則で表せ。(c) 受理する（非決定性）プッシュダウンオートマトンを構成せよ。(d) 上記のプッシュダウンオートマトンを模倣するプログラムを作成せよ（言語は何でも可）。

(1) $\{a^n b^n | n \geq 0\}$
(2) 偶数文字の回文
(3) a, b が同数現われる
(4) その他、特に正規言語でないような、<u>適当に非自明で興味深い</u>例を考えよ。

問 7. $\Sigma = \{a, b, +, -, \times, /, (,)\}$ から成る "文法に適っている" 数式を、

(1) 生成規則で表せ。
(2) 受理する（非決定性）プッシュダウンオートマトンを構成せよ。
(3) 上記のプッシュダウンオートマトンを模倣するプログラムを作成せよ。

但し、簡単のため、考える上で多少の制約・許容を行なって良い（$+, -$ を単項演算子としては使わないとか、普通なら付けない括弧が付いても気にしないとか）。

問 8. 例えばローマ字仮名変換のように、入力を 1 文字づつ読込みながら適宜出力をしてゆくような処理の定式化のために、"出力付き有限オートマトン" とでも言うべき**有限状態変換器 (finite state transducer)** と呼ばれる計算モデル（処理モデル）を考えよう。

(1) この計算モデルを定式化せよ。
(2) その下で、ローマ字仮名変換を実装してみよ（これを実行する "出力付き有限オートマトン" を構成せよ）。例えば、次のような制限下でも良い。
    - 入力文字集合 $\Sigma = \{\mathtt{a}, \mathtt{o}, \mathtt{k}, \mathtt{r}, \mathtt{y}\}$
    - 出力文字集合 $\Gamma = \{$ あ, お, か, こ, ら, ろ, や, よ, き, り, ゃ, ょ, っ$\}$
(3) ローマ字仮名変換を行なうプログラムを作成せよ。

問 9.　前問を発展させ、プッシュダウンスタックを持った有限状態変換器を考えよう。

(1) この計算モデルを定式化せよ。（実装を念頭に、決定性のモデルとしてもよい。）
(2) その下で、中置記法の式を入力して後置記法（逆ポーランド記法）に変換して出力するプッシュダウンスタック付き有限状態変換器を構成せよ。例えば、入力文字集合を $\Sigma = \{a, b, +, \times, (,)\}$ （ここに、a, b は変数名、演算子 $+, \times$ の優先順位は通常通り）とするような制限下でも良い。

問 10.　後置記法（逆ポーランド記法）で書かれた数式を入力して計算結果を返す "逆ポーランド電卓" を実装せよ。演算子は $+, -, \times, /$ などで、全て二項演算子としてよい。入力数値としては、

- 一桁の自然数 (0〜9) に限定（初級）
- 自然数に限定、二桁以上も可（中級）
- 実数に対応し、小数点付きの数値も受付ける（上級）

など適当な仕様を選べ。（尚、前問と繋げれば、中置記法の数式を計算する電卓にもなる。）

問 11.　文脈自由言語は、次の形の生成規則の幾つかで生成されることを示せ（**Chomsky の標準形**という）：
V を変数の集合、Σ を終端記号 (alphabet) の集合、$S \in V$ を初期変数として、

- $S \to \varepsilon$
- $A \to BC$ $(A \in V, B, C \in V \setminus \{S\})$
- $A \to a$ $(A \in V, a \in \Sigma)$

問 12.　文脈自由言語 L に対する Pumping Lemma（注入補題・反復補題）を考える。

(1) 文脈自由言語 L に対する Pumping Lemma を述べよ。
(2) L の生成規則を上問の Chomsky の標準形に取ったとする。長さ ℓ の語 $w \in L$ は生成規則を何回適用して得られるか。
(3) 語 $w \in L$ の構文解析木 T に於いて、w の各文字に至る道の長さが n 以下であるとき、w の長さ $|w|$ を上から評価せよ。
(4) L の生成規則の個数を N とする。語 $w \in L$ の構文解析木に於いて、どの文字に至る道でも同じ生成規則を高々一度しか用いないとするとき、w の長さ $|w|$ を上から評価せよ。
(5) 文脈自由言語 L に対する Pumping Lemma を示せ。但し、pumping 長 p を N を用いて明示的に与えよ。（ヒント：語 $w \in L$ の構文解析木に於いて、或る文字に至る道で同じ生成規則を 2 回使っていると、そこから Pumping Lemma の主張する語の分割が構成できる。）

問 13.　$\Sigma = \{a, b\}$ を alphabet とする次の言語について、(a) 文脈自由言語でないことを示せ（文脈自由言語に対する Pumping Lemma を用いよ）。(b) 文脈依存文法で記述することが出来るか。(c) 受理する（決定性または非決定性）チューリングマシンを構成せよ。(d) 上記のチューリングマシンを模倣するプログラムを作成せよ（言語は何でも可）。

(1) 同じ文字列 2 回の繰返しから成る文字列全体 $\{w^2 | w \in \Sigma^*\}$
(2) $\{a^n b^n a^n | n \geq 0\}$
(3) その他、特に文脈自由言語でないような、適当に非自明で興味深い例を考えよ。

問 14.　集合 X が可算 (enumerable) であるとは、自然数全体の集合 $\boldsymbol{N}$ との間に全単射 $X \to \boldsymbol{N}$ が存在することをいう。また、単射 $X \to \boldsymbol{N}$ が存在するとき、X が高々可算であると言う。

(1) 高々可算集合の高々可算個の合併集合が再び高々可算であることを示せ。
(2) 高々可算集合の有限部分集合全体の成す集合が再び高々可算であることを示せ。
(3) 可算集合の部分集合全体の成す集合（冪集合）が可算でないことを示せ。（ヒント：対角線論法）

問 15.　有限集合 Σ を alphabet とする言語で、チューリングマシンで認識できないものが存在することを、以下の手順で示せ。

(1) チューリングマシンの厳密な定式化を記述せよ。
(2) チューリングマシンの総数が可算個であることを示せ。
(3) 有限集合 Σ を alphabet とする言語の総数が可算でないことを示せ。
(4) 有限集合 Σ を alphabet とする言語で、チューリングマシンで認識できないものが存在することを示せ。