

問題解答例

[1章]

演習問題 1.1

1.

- (1) 空木 ϵ に対して, $V_t(\epsilon) = 0$ である.
- (2) 任意の記号 $a \in \Sigma$ および Σ 上の k 分木 $t_1, \dots, t_k$ に対して,
 $V_t(a(t_1, \dots, t_k)) = 1 + \sum_{i=1}^k V_t(t_i)$ である.

2.

- (1) 空木 ϵ に対して, $Lf(\epsilon) = 0$ である.
- (2) 任意の記号 $a \in \Sigma$ に対して, $Lf(a(\epsilon, \dots, \epsilon)) = 1$ である.
- (3) 任意の記号 $a \in \Sigma$ および Σ 上の k 分木 $t_1, \dots, t_k$ に対して, $t_1, \dots, t_k$ のうち少なくとも1つが空木でないならば, $Lf(a(t_1, \dots, t_k)) = \sum_{i=1}^k Lf(t_i)$ である.

3.

- (1) 空木 ϵ に対して, $N_a(\epsilon) = 0$ である.
- (2) Σ 上の k 分木 $t_1, \dots, t_k$ に対して, $N_a(a(t_1, \dots, t_k)) = 1 + \sum_{i=1}^k N_a(t_i)$ である.
- (3) a と異なる任意の記号 $b \in \Sigma$ および Σ 上の k 分木 $t_1, \dots, t_k$ に対して,
 $N_a(b(t_1, \dots, t_k)) = \sum_{i=1}^k N_a(t_i)$ である.

演習問題 1.2

1. 非終端記号としては, $\langle \text{定数} \rangle$, $\langle \text{数字列} \rangle$, $\langle \text{正数字} \rangle$, $\langle \text{数字} \rangle$ を用意し, 以下の生成規則によって文法を与えればよい. 開始記号は $\langle \text{定数} \rangle$ である.

- (1) $\langle \text{正数字} \rangle \rightarrow 1 \mid \dots \mid 9$
- (2) $\langle \text{数字} \rangle \rightarrow 0 \mid \langle \text{正数字} \rangle$
- (3) $\langle \text{数字列} \rangle \rightarrow \langle \text{数字} \rangle \langle \text{数字列} \rangle \mid \langle \text{数字} \rangle$
- (4) $\langle \text{定数} \rangle \rightarrow \langle \text{正数字} \rangle \langle \text{数字列} \rangle \mid \langle \text{数字} \rangle$

2. まず, 変数名を定義する文法には, 例題 1.4 で与えた以下の生成規則を用いられたい.

- (1) $\langle \text{変数名} \rangle \rightarrow \langle \text{英記号} \rangle \langle \text{記号列} \rangle \mid \langle \text{英記号} \rangle$
- (2) $\langle \text{記号列} \rangle \rightarrow \langle \text{記号} \rangle \langle \text{記号列} \rangle \mid \langle \text{記号} \rangle$
- (3) $\langle \text{英記号} \rangle \rightarrow a \mid \dots \mid z \mid A \mid \dots \mid Z$
- (4) $\langle \text{記号} \rangle \rightarrow 0 \mid \dots \mid 9 \mid \langle \text{英記号} \rangle$

これらに加えて、＜変数宣言＞と＜変数名リスト＞を非終端記号として用意し、以下の生成規則を加える。開始記号は＜変数宣言＞である。

- (5) ＜変数名リスト＞ $\rightarrow$ ＜変数名＞”,” ＜変数名リスト＞ | ＜変数名＞
- (6) ＜変数宣言＞ $\rightarrow$ Var “_” ＜変数名リスト＞

演習問題 1.3

1. 計算する関数を $VertexNum(t)$ とする。以下のように定義できる。

- (1) $VertexNum(t) = 0$ $t = \epsilon$ のとき
- (2) $VertexNum(t) = 1 + VertexNum(X) + VertexNum(Y)$
 $t = a(X, Y)$ と表されるとき

2. 計算する関数を $LeafNum(t)$ とする。以下のように定義できる。

- (1) $LeafNum(t) = 0$ $t = \epsilon$ のとき
- (2) $LeafNum(t) = 1$ $t = a(\epsilon, \epsilon)$ と表されるとき
- (3) $LeafNum(t) = LeafNum(X) + LeafNum(Y)$
少なくとも1つが空木でない X, Y に対して
 $t = a(X, Y)$ と表されるとき

演習問題 1.4

1. まず、基底ステップとして、 $n = 4$ の場合を考える。このとき、

$$4^2 = 16 = 2^4$$

であるから、 $n = 4$ のとき題意が成り立つ。次に、帰納ステップを証明する。まず、 $n = k$ ($k \geq 4$) のとき題意が成り立つと仮定する。このとき、

$$(k+1)^2 = k^2 + 2k + 1$$

である。ここで、帰納法の仮定から、

$$k^2 \leq 2^k$$

が成り立つので、

$$(k+1)^2 = k^2 + 2k + 1 \leq 2^k + 2k + 1$$

ここで、例題 1.6 より、 $2k + 1 < 2^k$ が $k \geq 3$ に対して成り立つので、

$$(k+1)^2 < 2^k + 2^k = 2^{k+1}$$

よって帰納ステップが成り立つ。

2. まず、基底ステップとして、 $n = 1$ の場合を考える。

$$4^1 - 1 = 3$$

であるので、 $n = 1$ のとき題意は成り立つ。次に、 $k \geq 1$ として、 $4^k - 1$ が 3 の倍数であると仮定する。このとき、

$$4^{k+1} - 1 = 4 \cdot 4^k - 1 = 4 \cdot (4^k - 1) + 3$$

となる。帰納法の仮定より、 $4^k - 1$ は 3 の倍数であるから、 $4^{k+1} - 1$ も 3 の倍数となる。よって、 $n = k + 1$ のときも題意が成り立つ。

演習問題 1.5

1. 関数 $f(n)$ が停止することを、 n に関する帰納法で示す。基底ステップとして、 $n = 0$ のときを考える。 $n = 0$ のときは、0 が返されて停止する。

次に帰納ステップを示す。まず、 $n = k$ ($k \geq 0$) のとき停止すると仮定する。 $f(k+1)$ は、 $k+1 \geq 1$ であるので、 $f(k)$ を再帰的に呼出す。帰納法の仮定より、 $f(k)$ は停止するので、 $f(k+1)$ も停止する。よって、帰納ステップも示された。

2. 2 分木の高さに関する帰納法で示す。高さが 0 のとき、つまり、空木のとき、頂点数は必然的に 0 となる。このとき、 $1 = 2^0 - 1$ であるので、題意が成り立つ。

次に帰納ステップを示す。高さ k ($k \geq 0$) の 2 分木の頂点数の最大値が $2^k - 1$ で与えられるものと仮定する。このとき、高さ $k+1$ の 2 分木で頂点数が最大のものを t とする。木 t が $t = a(t_1, t_2)$ と表されるとき、部分木 t_1, t_2 の高さは k 以下のはずである。なぜなら、高さ k を越えると、 t の高さが $k+1$ を越えてしまうからである。また、 t の頂点数が最大であるから、明かに t_1, t_2 の高さは k 未満であることはなく、 k となるはずである。さらに、 t_1, t_2 は高さ k の 2 分木の中で頂点数が最大のものであるはずである。よって、帰納法の仮定より、 t_1 と t_2 の頂点数は、 $2^k - 1$ であるので、 t の頂点数は $1 + (2^k - 1) + (2^k - 1) = 2^{k+1} - 1$ となる。よって、帰納ステップが示された。

[2章]

演習問題 2.1

1. 辺はその 2 つの端点を決めれば一意に定まるので、端点の選び方の個数を数えればよい。頂点数 n なので、 n 個の頂点から 2 個の頂点を選ぶ場合の数

は $\frac{n(n-1)}{2}$ となる. よって, $m \leq \frac{n(n-1)}{2}$ となる.

2. 例えば, 以下の G で与えられるグラフは自己補グラフである.

$$G = (V, E)$$

$$V = \{v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{\{v_1, v_2\}, \{v_2, v_3\}, \{v_3, v_4\}, \{v_4, v_5\}, \{v_5, v_1\}\}$$

3. G が頂点数 n の自己補グラフであるとする. G と G^c の辺集合の和集合は $\frac{n(n-1)}{2}$ 本の辺をもつ. G は自己補グラフであるから G と G^c の辺の本数は等しい. よって, $\frac{n(n-1)}{2}$ は偶数であり, ある整数 m を用いて $\frac{n(n-1)}{2} = 2m$ と表せる. よって, $n(n-1) = 4m$ より, n は 4 で割りきれるか, または, 4 で割ると 1 余る.

演習問題 2.2

1. 連結なグラフ G の任意の切断辺を $e = \{u, v\}$ とする. 辺 e がある全域木 T に含まれていないものとする. T は全域木であるから, T 上の u から v への道 α が唯一存在する. よって, G から e を取り除いても α が存在するので G の連結成分が増加することはない. これは, e が切断辺であることに矛盾する. よって, 辺 e はすべての全域木に含まれる.

逆に, 辺 e は G のすべての全域木に含まれるとする. このとき, e が切断辺でないとしたと, G から e を取り除いてできるグラフ G' は連結である. よって, 例題 2.1 により G' には全域木が存在する. G' の全域木は明らかに G の全域木でもあるので, e が G のすべての全域木に含まれることに矛盾する. よって, e は G の切断辺である.

2. 以下の図 1 の太線で与えられる.

3. グラフ G の頂点集合を V とし, $e = \{u, v\}$ とする. カット $C = \{\{u\}, V - \{u\}\}$ は, 辺集合 $\emptyset$ を侵害しない. また, e は重み最小なので, C に対する軽い辺である. よって, 定理 2.2 より, e を含む最小全域木が存在する.

4. グラフ G の頂点集合を V とし, $e = \{u, v\}$ を G のある最小全域木 T に含まれる辺とする. T から e を取り除くと, T は 2 つの連結成分に分割されるが, そのそれぞれの頂点集合を V_1, V_2 とする. このとき, カット $C = (V_1, V_2)$ を考えると, e は C の軽い辺である. なぜなら, もし, $w(e_*) < w(e)$ なる e_* が C と交差しているものとしたと, T から e を取り除いて e_* を加えてできる全域木は T より小さい重みをもつことになり, T が最小全域木であることに矛盾するからである. よって, e は C に対する軽い辺である.

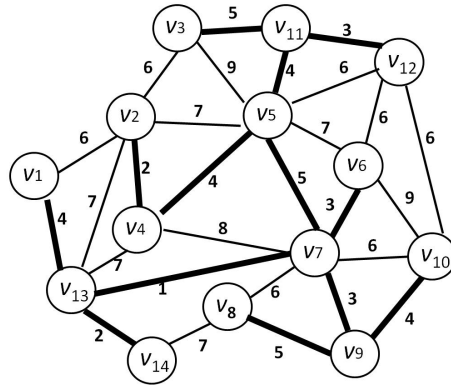


図 1 重みつきグラフ

5. グラフ G の最小全域木 T を考える. また G の w_{max} -最小全域木を T_* とし, T の最大の重みの辺を e とする. T から e を取り除くと, T は 2 つの連結成分に分割されるが, そのそれぞれの頂点集合を V_1, V_2 とする. このとき, カット $C = (V_1, V_2)$ を考えると, T_* は全域木なので, C と交差する辺 e_* をもつ. ここで, T が w_{max} -最小全域木でないと仮定すると, $w_{max}(T_*) < w(e)$ が成り立つ. よって, $w(e_*) \leq w_{max}(T_*) < w(e)$ となるので, T から e を取り除いて e_* を加えてできる全域木は T より小さい重みをもつことになり, T が最小全域木であることに矛盾する. よって, T は w_{max} -最小全域木である.

演習問題 2.3

1. $d_{i,j}^{(k)}$ によって, 中間頂点として $\{v_1, \dots, v_k\}$ を使うことを許された場合に頂点 v_i から v_j に正しくデータが送信される最大確率とする. すると, $d_{i,j}^{(k)} = \max\{d_{i,j}^{(k-1)}, d_{i,k}^{(k-1)} \times d_{k,j}^{(k-1)}\}$ によって $d_{i,j}^{(k)}$ ($i = 1, \dots, n$) を求めることができる. ただし, $k = 0$ の場合は, v_i から v_j に辺がある場合は, $d_{i,j}^{(k)} = w(v_i, v_j)$ となり, v_i から v_j に辺がない場合は, $d_{i,j}^{(k)} = 0$ となる.

[3章]

演習問題 3.1

1. M の動作を説明する. M は初期状態 p_0 から始める. 入力記号 a をひとつ読み込む度にそれを消去し, かつ状態 p_1 に移りヘッドを右へ移動していく. そして b と出逢ったとき状態を p_2 に変えて b の個数 (n 個) だけ (b を) c に代え

て最後部に**移動**させ**複製**を作る．このときテープ上は“ $\#a^{m-1}b^nc^n\#$ ”となっている．そして状態を p_3 に変えてヘッドを左端へ移動させ、 $\#$ と出逢ったとき p_0 に戻る．以上の動作をすべての a が無くなるまで繰り返す．このときテープ上は“ $\#b^nc^{mn}\#$ ”となっているので、あとは b^n を消去するだけでよい．

2. M は初期状態 p_0 から始める．入力記号 a (または b) をひとつ読み込む度にそれを消去し、かつ状態 p_a (あるいは p_b) に移りヘッドを右へ移動していく．そして c と出逢ったとき直後の記号を調べ、それが a (あるいは b) のときのみそれを消去して、状態を p_1 に変えてヘッドを左端へ移動し空白記号に出会ったとき状態を p_0 に戻す．以上の動作を繰り返す．状態 p_0 で記号 c を読んだとき、直後の記号が空白のとき、そのときにかぎり最終状態 p_f に移り、停止する．

3. M は3-テープチューリングマシンであり初期状態 p_0 から始める．第1テープ上には入力記号列 a^n を保持する．第2テープ上では b を生成し、以後第3テープを作業領域として $bbb, b^5, b^7, \dots$, のように奇数個の b からなる記号列を次々と生成し、それを第2テープ上の記号列の後尾にコピーし接続していく．この接続を行う毎に、第1テープ上の a^n と第2テープ上の b^m とを比較し、 $m = n$ と判明したとき、最終状態にはいり入力列 a^n を受理し停止する．第2テープ上では常に $m = 1 + 3 + 5 + \dots + (2k - 1) = k^2$ ($k \geq 1$) 個の b が生成されるので、受理される入力列は a^{k^2} ($k \geq 1$) である．

4. 与えられた非決定性チューリングマシン M において、次動作の選択数の最大値を m とすると、どのような有限長の動作系列もアルファベット $\{1, 2, \dots, m\}$ 上の有限長の記号列に対応している．(もちろん、すべての有限長の記号列が動作系列を表しているわけではないから逆は必ずしもいえない．) M を模倣する3テープチューリングマシン M' を図2のように構成する． M' は第1のテープ上に M の入力テープを保存し、第3テープ上にそのコピーを作る．さらに、第2のテープ上で $\{1, 2, \dots, m\}$ 上の有限長の記号列 s を(それを m 進数として)小さい順に次々と生成してゆき、各 s に対してコピーされた入力に対する M による動作系列 s を、第3のテープ上で模倣する．もし M が最終状態に入るならば、そのような動作系列が存在し、第2テープ上で必ず生成されるから、その時 M' も最終状態に入り受理する．もし M が最終状態へ入るような動作系列が存在しないのならば、 M' も受理しないことになる．

演習問題 3.2

1. (有限オートマトン $\implies$ 正則文法) 与えられた有限オートマトン $A = (Q, \Sigma, \delta, p_0, F)$ (ここで $Q \cap \Sigma = \emptyset$ と仮定してよい) に対して正則文法 $G = (Q, \Sigma, P, p_0)$ を以下のように構成する: $P = \{p \rightarrow aq \mid \delta(p, a) = q\} \cup \{p_0 \rightarrow \varepsilon\}$ ($p_0 \in F$ の場合のみ)．このとき $\delta(p_0, w) \in F \iff p_0 \Rightarrow^* w$ が ($|w| = n$ に関する帰納法により) 示せる．よって $L(A) = L(G)$ を得る．

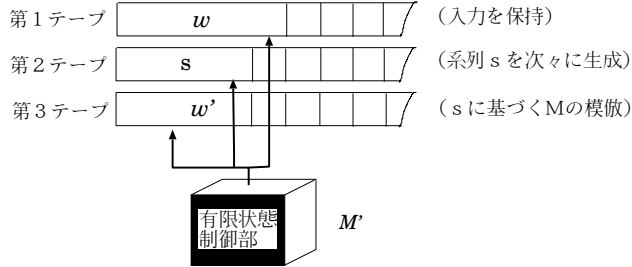


図 2 決定性 TM: M' による非決定性 TM: M の模倣

(正則文法 $\Rightarrow$ 有限オートマトン) 与えられた正則文法 $G = (N, \Sigma, P, S)$ に対して (非決定性) 有限オートマトン $A = (N \cup \{q_f\}, \Sigma, \delta, S, \{q_f\})$ を以下のように構成する: $\delta(X, a) \ni Y$ ($X \rightarrow aY \in P$ のとき), $\delta(X, a) = q_f$ ($X \rightarrow a \in P$ のとき). このとき $S \Rightarrow^* w \iff \delta(S, w) = q_f$ が $(|w| = n$ に関する帰納法により) 示せる. よって $L(G) = L(A)$ を得る. (非決定性有限オートマトンはそれと等価な (決定性) 有限オートマトンに変換できるので証明は完了する.)

2. 準備として, 任意のプッシュダウンオートマトンは空スタック受理によって言語を受理できるように変換可能であることに留意せよ.

(プッシュダウンオートマトン $\Rightarrow$ 文脈自由文法) 与えられたプッシュダウンオートマトン $M = (Q, \Gamma, \Sigma, \delta, p_0, Z_0, \emptyset)$ (ここで Q, Γ, Σ は互いに共通部分がないと仮定してよい) に対して文脈自由文法 $G = (N, \Sigma, P, S)$ を以下のように構成する: $N = \{S\} \cup \{[p, a, X] \mid p \in Q, a \in \Sigma, X \in \Gamma\}$ (3つ組 $[p, a, X]$ が一つの非終端記号である). P は以下の規則からなる:

- 1) すべての $q \in Q$ に対して $S \rightarrow [p_0, Z_0, q]$ は P の規則.
- 2) $\delta(p, a, X) \ni (q_1, Y_1 \cdots Y_n)$ のとき, すべての重複を許した可能な組合せ $q_2, q_3, \dots, q_{n+1} \in Q$ に対して,
 $[p, X, q_{n+1}] \rightarrow a[q_1, Y_1, q_2][q_2, Y_2, q_3] \cdots [q_n, Y_n, q_{n+1}]$ は P の規則.
- 3) $\delta(p, a, X) \ni (q_1, \epsilon)$ のとき, $[p, X, q_1] \rightarrow a$ は P の規則.

このとき $(p_0, w, Z_0) \vdash (q, \epsilon, \epsilon) \iff S \Rightarrow [p_0, Z_0, q] \Rightarrow^* w$ が示せる. よって $L(M) = L(G)$ を得る.

準備として, 任意の文脈自由言語 L は, その規則が「 $A \rightarrow a\alpha$ (ここで $a \in \Sigma, A \in N, \alpha \in N^*$)」の形のみである **Greibach 標準形**とよばれる文脈自由文法 G によって生成されることに留意せよ. ただし, G は $\epsilon \in L$ のとき

のみ $S \rightarrow \varepsilon$ を許し、他の規則の右辺は S を含まないとする。

(文脈自由文法 $\Rightarrow$ プッシュダウンオートマトン) 与えられた Greibach 標準形の文脈自由文法 $G = (N, \Sigma, P, S)$ に対してプッシュダウンオートマトン $M = (\{q\}, N, \Sigma, \delta, q, S, \emptyset)$ を以下のように構成する: $P = \{X \rightarrow a\alpha \mid X \in N, a \in \Sigma \cup \{\varepsilon\}, \alpha \in N^*\}$ (ここで $a = \varepsilon$ ときは $X = S$ かつ $\alpha = \varepsilon$ である)。このとき $S \Rightarrow^* w \iff (q, w, S) \vdash^* (q, \varepsilon, \varepsilon)$ が示せる (G においては最左導出, すなわち最も左に出現する非終端記号を常に書き換える導出を考えるとよい)。よって $L(G) = L(M)$ を得る。

3. (チューリングマシン $\Rightarrow$ 形式文法) 与えられたチューリングマシン $M = (Q, \Gamma, \Sigma, \delta, q_0, \bar{b}, F)$ に対して形式文法 $G = (N, \Sigma, P, S)$ を以下のように構成する: (ここで $\Sigma' = \Sigma \cup \{\varepsilon\}$ とする)

$$1) N = Q \cup \{S, S_1, S_2\} \cup \{[a, X] \mid a \in \Sigma', X \in \Gamma\}$$

$$2) P = \{S \rightarrow q_0 S_1\} \cup \{S_1 \rightarrow [a, a] S_1 \mid a \in \Sigma\} \quad (43)$$

$$\cup \{S_1 \rightarrow S_2\} \cup \{S_2 \rightarrow [\varepsilon, \bar{b}] S_2\} \cup \{S_2 \rightarrow \varepsilon\} \quad (44)$$

$$\cup \{p[a, X] \rightarrow [a, Y] q \mid a \in \Sigma', \delta(p, X) = (q, Y, R)\} \quad (45)$$

$$\cup \{[a, Z] p[b, X] \rightarrow q[a, Z][b, Y] \mid a, b \in \Sigma', Z \in \Gamma, \delta(p, X) = (q, Y, L)\} \quad (46)$$

$$\cup \{[a, X] p_f \rightarrow p_f a p_f, p_f[a, X] \rightarrow p_f a p_f \mid a \in \Sigma', X \in \Gamma, p_f \in F\} \quad (47)$$

$$\cup \{p_f \rightarrow \varepsilon\}. \quad (48)$$

入力列 $w = a_1 \cdots a_n (\in \Sigma^*)$ に対して、 G は w のコピーと M が w の認識作業に必要とする (であろう) 領域 k を (非決定的に推測して) 確保すべく規則 (43) と (44) を用いて

$$S \Rightarrow^* q_0[a_1, a_1] \cdots [a_n, a_n][\varepsilon, \bar{b}]^k$$

なる導出を行う。さらに規則 (45) と (46) を用いると、

$$q_0[a_1, a_1] \cdots [a_n, a_n][\varepsilon, \bar{b}]^k \Rightarrow^* [a_1, X_1] \cdots [a_{t-1}, X_{t-1}] p_f[a_t, X_t] \cdots [a_{n+k}, X_{n+k}]$$

なる導出が得られる。(ここで $i = n+1$ 以降の a_i は ε であり、後ろ側の幾つかの X_j はブランク記号 $\bar{b}$ であることに留意。) 最後に、規則 (47) と (48) を用いると、

$$[a_1, X_1] \cdots [a_{t-1}, X_{t-1}] p_f[a_t, X_t] \cdots [a_{n+k}, X_{n+k}] \Rightarrow^* a_1 a_2 \cdots a_n$$

が可能である。よって $L(M) \subseteq L(G)$ がいえる。逆に $w \in L(G)$ であれば、そ

これは規則 (47) と (48) を用いていることになり、これは w が M を最終状態へ導くことを意味する。よって $L(G) \subseteq L(M)$ でもあり、結局 $L(M) = L(G)$ がいえる。

(形式文法 $\Rightarrow$ チューリングマシン) これはいわゆる「チャーチの仮説」を用いると、与えられた形式文法 G において終端記号列 w を生成する過程は『機械的に実行可能な操作の有限系列』であることから明らかにチューリングマシンによって実現できる。よって、 $L(G) = L(M)$ となるチューリングマシン M が構成可能である。(より詳細な M の説明としては以下のようなもので十分であろう。)

入力列 $w(\in \Sigma^*)$ が与えられたとき、 M は「 w をテープ上に保持し、かつ G における導出過程を模倣する」ことを続ける。すなわち、 G の導出: $\beta\alpha\beta' \Rightarrow \beta\gamma\beta'$ (ここで規則 $\alpha \rightarrow \gamma$ が用いられたとする) を模倣するのに、 M はそのテープ上で、 $\beta\alpha\beta'\#w$ のような記号列を保持する (記号 $\#$ は区切り)。そして、左から操作していった非決定的に部分列 α を見つけて、それを γ で置き換える。このとき M のテープは $\beta\gamma\beta'\#w$ となっている。そこで $\beta\gamma\beta' = w$ であるかどうかを照合し、もし YES ならば、最終状態へ入って w を受理する。もし NO ならばさらにこの“模倣”操作を繰り返す。 $w \in L(G)$ ならば w の G による導出が存在するので、 M はそれを模倣でき、いつかは YES となる。よって $L(G) = L(M)$ となる。

演習問題 3.3

1. 図 3 はバッファ容量 $b = 3$ の場合の初期状態を示す。
2. 図 4 のペトリネットは、 $C_1, C_2, C_3, M_1, M_2, M_3$ に 1 個ずつのトークンを

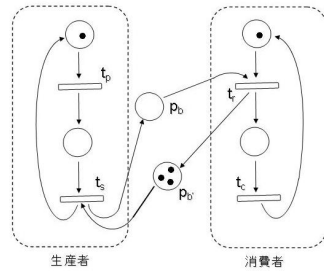


図 3 生産者/消費者問題のペトリネットによるモデリング：容量 $b = 3$ の場合

もつ初期マーキング μ_0 から，哲学者 1 が冥想状態 M_1 から 2 つの箸 C_1, C_3 を用いて食事状態 D_1 になった状況が表されている。

3. 図 5 の p_s と p_f は各々与えられたプログラムの初期状態と最終状態に対応

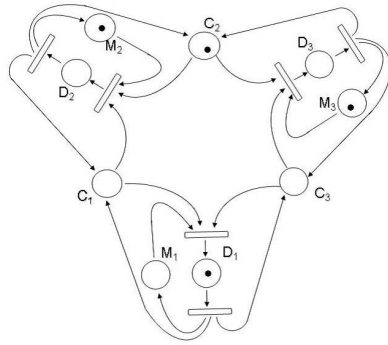


図 4 食事をする哲学者問題のペトリネットによるモデリング

するプレースを表す。

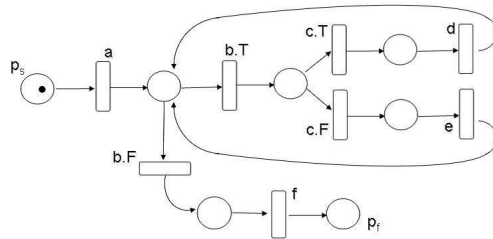


図 5 $3x+1$ 問題のペトリネットによるモデリング

演習問題 3.4

1. 図 3.6(a) のペトリネットを PN_1 とするとこれは活性でない．一方，図 6(a) のペトリネット PN_2 は活性である．このとき，これらはともに図 6(b) にある同一の可達木をもつ．

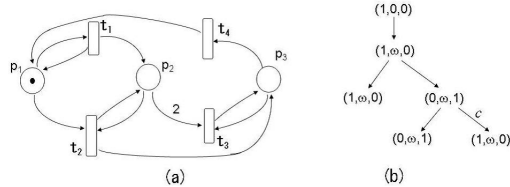


図6 ペトリネット (a) PN_2 と (b) その可達木

2. ペトリネットの可達木 T に ω が現れるとする. このとき ω に対応するプレースのトークン数は (ω の性質から) 任意に増大し得る. すなわち, ペトリネットは有界でない. 逆に, ペトリネットが有界でないとすると, 可達なマーキングの個数は無限である. 可達木 T を有限的に表すためには ω を導入する必要があり, T は必然的に ω を含む.

3. 無限の可達木をもつペトリネットと初期マーキングの例が図7に与えられる.

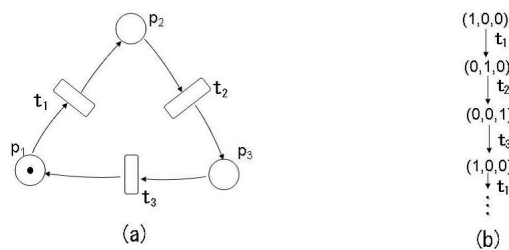


図 7 無限の可達木をもつペトリネットと初期マーキングの例

演習問題 3.5

1. (1).

$$M_{in} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 2 & 0 \\ 0 & 2 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad M = \begin{bmatrix} 0 & 2 & 0 \\ -1 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & -1 & -1 \end{bmatrix}$$

(2).

$$t_1 : (1, 0, 0, 0)M_{in} = (1, 0, 0) \not\leq \mu = (0, 1, 1)$$

$$t_2 : (0, 1, 0, 0)M_{in} = (1, 1, 0) \not\leq \mu$$

$$t_3 : (0, 0, 1, 0)M_{in} = (0, 2, 1) \not\leq \mu$$

$$t_4 : (0, 0, 0, 1)M_{in} = (0, 1, 1) \leq \mu$$

よって t_4 のみが発火可能なトランジションである。

(3). 状態方程式 $\mu' - \mu = xM$ において $x = (x_1, x_2, x_3, x_4)$ とおくと,
 $(2, 0, 0) - (0, 1, 1) = xM$, すなわち

$$(2, -1, -1) = (x_1, x_2, x_3, x_4) \begin{bmatrix} 0 & 2 & 0 \\ -1 & 0 & 1 \\ 0 & -2 & 0 \\ 1 & -1 & -1 \end{bmatrix}. \text{ これより得られる}$$

$$\text{連立方程式} \begin{cases} -x_2 + x_4 = 2 \\ 2x_1 - 2x_3 - x_4 = -1 \\ x_2 - x_4 = -1 \end{cases} \text{ は解をもたない. よって定理 3.1 よ}$$

り可達でない.

[補足：ペトリネットの標準形について] 4項組 (PN, μ_0, F, f) (ここで $PN = (P, T, A)$) は次のような性質をもつとき**標準形**であるという.

(1). P は唯一のプレース p_s (開始プレース) を持ち, 「 μ_0 は開始プレースにのみ 1 個のトークンをもち, これ以外はゼロ」である. また p_s を出力プレースとするようなトランジションは T にない.

(2). P は唯一のプレース p_f (最終プレース) を持ち, p_f を入力プレースとするようなトランジションは T にない. また p_f にトークンをもち, かつ発火不可能な (μ_0 から) 可達なマーキングが存在する.

PN の計算は $\mu_0(p_s) = 1$ かつ $\mu_0(p_f) = 0$ なる初期マーキング μ_0 で始まり, $\mu(p_f) = 1$ なる最終マーキング μ で終る.

事実として、『任意のペトリネットに対してそれと等価な標準形ペトリネットが存在する』ことが知られている (例えば, 文献 13) を参照). 図 8 (c) にはテキストの図 3.6 にある PN と等価な標準形のペトリネットが与えられている. 図中において, 実線で描かれた部分 (元の PN) に破線のアークと灰色で描かれたトランジションとプレースが追加されて標準形を構成している.

2.(1). (略証) ペトリネット言語 L_0 に対して 4 項組 (PN_0, μ_0, F_0, f_0) によって $L_0 = L(PN_0, \mu_0, F_0, f_0)$ であるとする. ここで PN_0 は開始プレース p_s と最終プレース p_f をもつ (上記の) 標準形であると仮定してよい. 与えられた $x, y \in \Sigma^*$ に対して, $\{x\}$ と $\{y\}$ を各々規定するペトリネット PN_x と PN_y は容易に構成できる ($x = a, y = ab$ のときの構成が各々図 8(a),(b) に与えられ

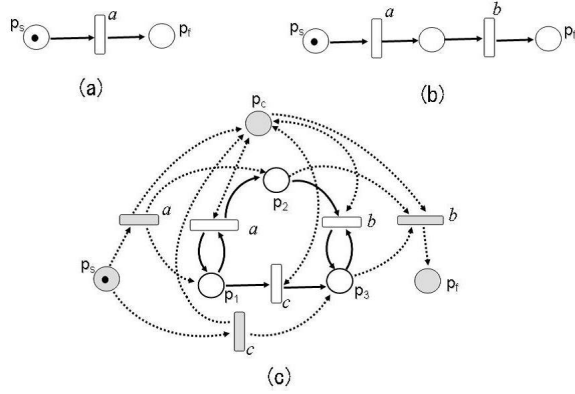


図 8 $\{a^i cb^i \mid i \geq 0\}$ を規定する標準形のペトリネット

ている)。このとき、言語 $L = \{x^i wy^i \mid w \in L_0, i \geq 1\}$ を生成する 4 項組 (PN, μ, F, f) を次のように構成する：

まず新たにプレース p_c を導入して、 PN_x が一度実行される度にトークンを 1 つずつ p_c に保持する（つまり x^i の回数 i をカウントする）。そして、 PN_0 の実行を終えた後、今度は p_c にあるトークンを 1 つずつ消費して PN_y を一度ずつ実行する。これにより最終マーキング F は唯一の $(0, 0, \dots, 1)$ からなる。（ f は f_0 と恒等関数とから自然に決まる関数とする。）

図 9 は、 $x = a, y = ab, L_0 = \{a^n cb^n \mid n \geq 0\}$ のとき構成すべきペトリネット PN が示されている。

(2). ペトリネット言語の族は正則言語族を含む (定理 3.2) ので、すべての有限集合を含む。言語 $L_i = L(PN_i, \mu_i, F_i, f_i) (i = 1, 2)$ とする。ここで PN_1 と PN_2 は共通の記号を持たないとしてよい。このとき、 $L_1 \cup L_2 = L(PN_3, \mu_3, F_3, f_3)$ となる 4 項組 (PN_3, μ_3, F_3, f_3) は $PN_3 = PN_1 \cup PN_2$, $\mu_3 = (u_1, \dots, u_p, v_1, \dots, v_q) = (\mu_1; \mu_2)$ （ここで $\mu_1 = (u_1, \dots, u_p)$, $\mu_2 = (v_1, \dots, v_q)$ ）, $F_3 = \{(u; \mu_2), (\mu_1; v) \mid u \in F_1, v \in F_2\}$, $f_3 = f_1 \cup f_2$ と定義することにより得られる。また、 $L_1 L_2 = L(PN_4, \mu_4, F_4, f_4)$ となる 4 項組 (PN_4, μ_4, F_4, f_4) は「 PN_1 において $t \in T$ が最終マーキングに至る最後のトランジションであるとき、 t から μu_2 の開始プレースへアークを張る」ように修正したものを PN_4 とし、 $\mu_4 = \mu_2, F_4 = F_2, f_4 = f_1 \cup f_2$ とすればよい。

以上によりペトリネット言語族は和集合・連接に関して閉じており、性質 (1)

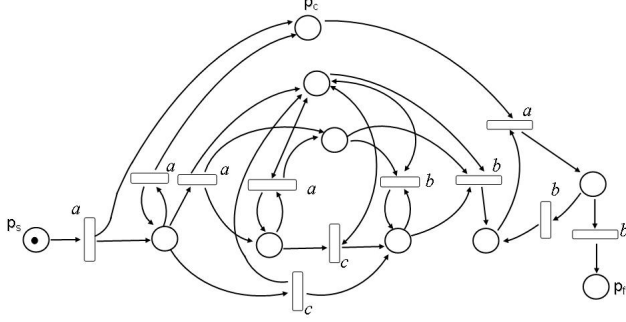


図 9 $\{a^i w b^i \mid w \in L_0, i \geq 1\}$ を生成するペトリネットと初期マーキングの例

を満たすので限定文脈自由言語族を含んでいる。

3. (略証) 言語 $L_{mir} = \{ww^R \mid w \in \{a, b\}^*\}$ を生成する 4 項組 (PN, μ_0, F, f) が存在すると仮定しよう。 $ww^R \in L_{mir}$ (ここで $k = |w|$ とする) を認識するペトリネット PN は前半の w を実行した段階で 2^k 種類の w を記憶する必要がある、したがって k 回推移した後で 2^k 個の異なる可達なマーキングを実現する必要がある。 さもないと、 $|w_1| = |w_2| = k$ かつ $w_1 \neq w_2$ なる w_1 と w_2 に対して $\delta(\mu_0, w_1) = \delta(\mu_0, w_2)$ である。 このとき

$$\delta(\mu_0, w_1 w_2^R) = \delta(\delta(\mu_0, w_1), w_2^R) = \delta(\delta(\mu_0, w_2), w_2^R) = \delta(\mu_0, w_2 w_2^R) \in F$$

であるから、ペトリネットは $w_1 w_2^R$ を認識することになり矛盾する。 一方、発火系列 $\sigma = t_1 t_2 \cdots t_k$ により μ_0 から可達なマーキングの種類は、発火ベクトルを $\phi(\sigma) = (v_1, \dots, v_m)$ (m は PN のトランジションの個数) とすると「整数 k の m 個の分割の総数」に等しく、 ${}_{k+m-1}C_{m-1}$ であることが知られている。 よって、

$${}_{k+m-1}C_{m-1} < (k+m)^m < 2^k$$

となる十分大きな k を考えたとき、可能な可達マーキングの種類が 2^k に達することは不可能になる。 よって L_{mir} はペトリネット言語ではない。

[4章]

演習問題 4.1

任意の $x \in X$ を固定して考える. すると任意の $f \in B^X$ に対して $f(x) \in \{0, 1\}$ が定まる. この意味で f を $B = \{0, 1\}$ 上のブール変数とみなすと, 演算 $\{+, \cdot, '\}$ の定義より例 4.1(1) の B_1 に対するときと同じ論法で B_f は $\mathcal{A}$ の性質を満たすことが示せる. よってブール代数である.

演習問題 4.2

(等式 $\alpha = \beta$ に含まれる演算の個数に関する帰納法によって示すのが通常であるが, ここでは別法を示す.)

演習問題 4.6 の (1) より, 任意のブール多項式は 0, 1, 演算 “'” と多数決関数 $f(x, y, z)$ によって表すことができる. よって α, β をそのような表現であるとし, $\alpha = \beta$ と仮定する. (このとき $x + y = f(x, y, 1)$, $xy = f(x, y, 0)$ なる関係を用いていることに留意せよ.) この等式 $\alpha = \beta$ に現れている 0 を 1 に 1 を 0 に置き換えたとすると, 多数決関数は自己双対ゆえ不変であり得られる式は $\alpha = \beta$ の双対式, すなわち $\alpha^* = \beta^*$ となる.

逆に, $\alpha^* = \beta^*$ ならば $\alpha = (\alpha^*)^* = (\beta^*)^* = \beta$ である.

演習問題 4.3

(P9): $x + x'y = (x + x')(x + y) = 1(x + y) = x + y$. もうひとつの $x(x' + y) = xy$ は第 1 式と双対原理による.

(P10): $xz + x'y + yz = xz + x'y + yz1 = xz + x'y + yz(x + x')$
 $= xz + x'y + yzx + yzx' = (xz + xzy) + (x'y + x'yz)$
 $= xz + x'y$

もうひとつの式は第 1 式と双対原理による.

(P11): $(xy' + x'y)' = (xy')'(x'y)' = (x' + y)(x + y')$
 $= xx' + x'y' + yx + yy' = 0 + xy + x'y' + 0$
 $= xy + x'y'$

(P12): $(x + y)(xy)' = (x + y)(x' + y') = xx' + xy' + x'y + yy'$
 $= xy' + x'y.$

演習問題 4.4

題意と定義より束 B は (A1), (A2), (A4), (A5), (A6) を満たす. よって B は (A3) 分配律を満たすことを示せばよい. 定義より任意の x, y に対して「 $x + x'y = x + y$ 」および「 $1 + x = 1$ 」が成り立つことに留意せよ.

$u = xy + xz$ および $v = x(y + z)$ とする. このとき, $u + v' = 1$ かつ $uv' = 0$ を示せば, (P6: 補元の一意性) より $u = v$ が成り立つ. ($uv' = 0$ は $u + v' = 1$

と双対原理より導かれるので) 以下では $u + v' = 1$ を示せばよい.

$$\begin{aligned} u + v' &= (xy + xz) + (x(y + z))' = xy + xz + (x' + (y + z)') \\ &= (xy + x') + xz + y'z' = (x' + xz) + (y + y'z') \\ &= (x' + z) + (y + z') = (x' + y) + (z + z') = (x' + y) + 1 = 1 \end{aligned}$$

よって $xy + xz = x(y + z)$ が成り立つ. もうひとつの分配律は双対原理による.

演習問題 4.5

$$\begin{aligned} g_1(x, y, z) &= xyz + xy'z' + xy'z + x'y'z + xy'z' + x'y'z' \\ &= xy(z + z') + xy'(z + z') + x'y'(z + z') \\ &= xy + (x + x')y' = xy + y' = x + y' \cdots (P9) \text{ より} \\ g_2(x, y, z) &= xyz + xy'z + x'yz + x'yz' + xyz + x'y'z \\ &= xz(y + y') + x'y(z + z') + yz(x + x') \\ &= xz + x'y + yz = xz + x'y \cdots (P10) \text{ より} \end{aligned}$$

演習問題 4.6

1. 演算 $+$ と $\cdot$ の各々が多数決関数によって表せることを示せば十分である. 実際, $f(x, y, z) = (x + y)(y + z)(z + x) = xy + yz + zx$ に留意すると

$$\begin{cases} x + y &= (x + y) \cdot 1 \cdot 1 = (x + y)(y + 1)(1 + x) = f(x, y, 1) \\ xy &= xy + 0 + 0 = xy + y \cdot 0 + 0 \cdot z = f(x, y, 0). \end{cases}$$

2. まず 1. よりブール関数 $f(x_1, \dots, x_n)$ は演算 $'$ と多数決関数によって表されていると仮定してよい. 1. の証明から分かるように, $f(x, y, 0)$ と $f(x, y, 1)$ の形の多数決関数では, 0 と 1 の置き換えは “+” と “ $\cdot$ ” のそれに対応する. また, 通常の多数決関数 $f(x, y, z)$ に関しては自己双対であることから不変となることに留意せよ.

($\Leftarrow$) ブール関数 $f(x_1, \dots, x_n)$ が 0 と 1 を含まずに演算 $'$ と多数決関数だけによって表されているならば, その双対はもとの関数と等しいので, 自己双対である.

($\Rightarrow$) もし $f(x_1, \dots, x_n)$ に 0 あるいは 1 が含まれているとする. (f は $'$ と多数決関数によって表されているという仮定より) f における 0 と 1 を入れ換えるだけで f の双対 f^* が得られる. f^* と f との違いは 0 が 1 (1 が 0) に入れ換わっているだけであり, 他の部分は等しい. このとき f は自己双対 (すなわち $f = f^*$) であることから, $0 = 1$ でなければならず矛盾を生じる. よって f

には 0 も 1 も含まれない.

演習問題 4.7

まず $x' = (x + x)' = x \downarrow x$ である. これより, 演算 “ $|$ ” は

$$x|y = (xy)' = (x' + y')' = (x' \downarrow y') = ((x \downarrow x) \downarrow (y \downarrow y))$$

のように NOR だけで表せる. よって定理 4.6 により任意の $f \in \mathcal{F}_n$ は演算 NOR だけを用いて表現できる.

演習問題 4.8

$f(x, y, z)$, $g(x, y, z, u)$, $g(x, y, z, u, v)$ に関するカルノー図は図 10 の (a), (b), (c) のようになる.

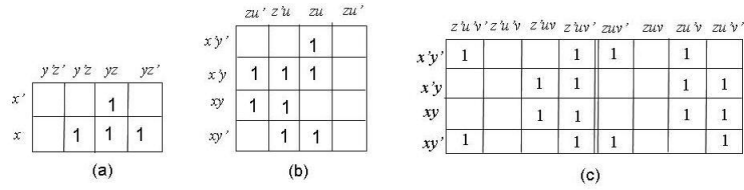


図 10

演習問題 4.9

1. および 2. に与えられたブール関数の各積項に番号を以下のようにつける:

$$f(x, v, y, z) = (1)xvyz + (2)xvyz' + (3)xvy'z + (4)xv'y'z + (5)x'vy'z + (6)x'v'yz + (7)xv'y'z' + (8)x'v'y'z'$$

$$g(x, v, y, z) = (1)xvyz + (2)x'vyz + (3)xvy'z + (4)xv'y'z + (5)x'vy'z + (6)xv'y'z' + (7)x'vy'z' + (8)x'v'y'z'$$

f と g に関する「主項と積項からなる行列」を求めると表 1 のようになる. これより, 各々の最簡形は

$$f(x, v, y, z) = yz' + xvy + xv'y' + x'v'z \text{ および } g(x, v, y, z) = vz + xv'z' + x'y'z' \text{ となる.}$$

表 1：主項と積項からなる行列

f	1	2	3	4	5	6	7	8
$\{3, 4, 5, 8\}$ $y'z$			x	x	x			x
$\{4, 7\}$ $xv'y'$				x			x	
$\{6, 8\}$ $x'v'z$						x		x
$\{1, 3\}$ xvz	x		x					
$\{1, 2\}$ xvy	x	x						

g	1	2	3	4	5	6	7	8
$\{1, 2, 3, 5\}$ vz	x	x	x		x			
$\{4, 6\}$ $xv'z'$				x		x		
$\{5, 7\}$ $x'vy'$					x		x	
$\{6, 8\}$ $v'y'z'$						x		x
$\{7, 8\}$ $x'y'z'$							x	x

演習問題 4.10

題意を満たす自動販売機を状態遷移図は図 11(a) のようになる。ここで、状態遷移のラベルにおける “/” で区切られた左側（入力記号）の 0 と 1 は “何も入力しない” と “100 円コインの入力” を表し、右側（出力記号）の 0 と 1 は “何も出さない” と “300 円の品物を出す” を表す。3つの状態の割当ては二つの D-FF を用いることにして、各々からの出力変数を A_1 , A_2 で表す。これより状態割当てを施した後の状態遷移表と書き直した状態遷移表は図 11(b) のようになる。ここで、変数 X は入力を、変数 F は出力を表す。これより

$$\begin{aligned}
 F(t_{n+1}) &= X(t_n)A_1(t_n)A'_2(t_n) \\
 A_1(t_{n+1}) &= X'(t_n)A_1(t_n)A'_2(t_n) + X(t_n)A'_1(t_n)A_2(t_n) \\
 A_2(t_{n+1}) &= X'(t_n)A'_1(t_n)A_2(t_n) + X(t_n)A'_1(t_n)A'_2(t_n) \\
 &= A'_1(t_n)(X(t_n) \oplus A'_2(t_n))
 \end{aligned}$$

と計算できる。よって F を実現する順序回路は図 12 のようになる。

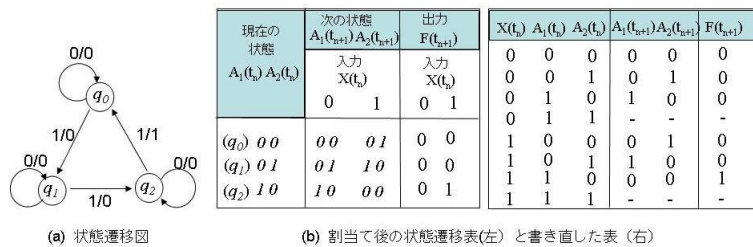


図 11

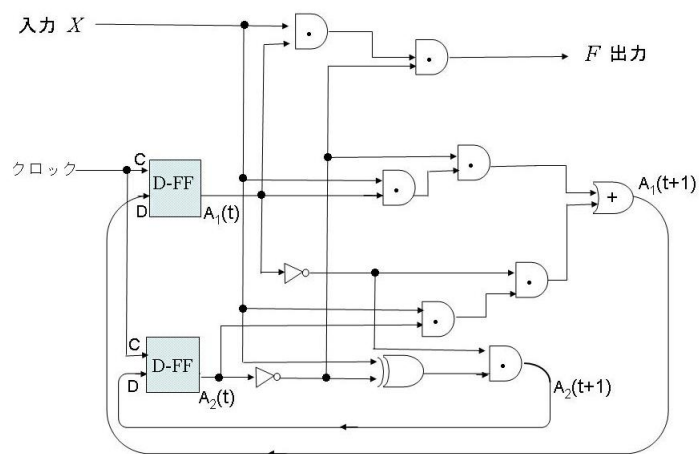


図 12

[5章]

演習問題 5.1

- $mx \equiv n \pmod{k}$ かつ $my \equiv n \pmod{k}$ とすると, $mx \equiv n \equiv my \pmod{k}$. このとき $(m, k) = 1$ であるから命題 5.1 の (3) より $x \equiv y \pmod{k}$ となる.
- $x_0 = 10600$, $x_1 = 11$ において, 拡張ユークリッドの互除法によってアルゴリズム Ex-EUCLID(10600, 11) を実行する. その結果,

i	$x_{i-1} = q_i + r \times x_i$	(m, n)	A_1 の値	B_1 の値
0			A	B
1	$10600 = 963 \times 11 + 7$		B_2	$A_2 - 963B_2$
2	$11 = 1 \times 7 + 4$		$A_3 - B_3$	$-963A_3 + 964B_3$
3	$7 = 1 \times 4 + 3$		$-A_4 + 2B_4$	$964A_4 - 1927B_4$
4	$4 = 1 \times 3 + 1$		$2A_5 - 3B_5$	$-1927A_5 + 2891B_5$
5	$3 = 3 \times 1 + 0$	1	$-3(A_5 = 0, B_5 = 1)$	$2891(A_5 = 0, B_5 = 1)$

となり $\langle -3, 2891, 1 \rangle$ を出力する。よって求める整数解は $(x, y) = (2891, -3)$ となる。 $(11, 10600) = 1$ であるから求める唯一の解は $x \equiv 2891 \pmod{10600}$ となる。

演習問題 5.2

- C_1 におけるどの二つの符号のハミング距離も 2 である。よって C_1 は 1 重誤り検出可能な符号であるが、誤り訂正可能な符号ではない。
- C_2 の最小ハミング距離は 4 であることがわかる。よって C_2 は 3 重誤り検出符号であり、かつ 1 重誤り訂正符号である。

演習問題 5.3

- ハミング符号の定義よりパリティ検査行列 H の列ベクトル $\mathbf{h}_i$ はすべて異なる。 F_2 において異なる値の和は 1 であるから、任意の 2 つの列ベクトルの和は $\mathbf{0}$ にならない。よって、任意の 2 つの列ベクトルは一次独立である。
- (5.19) 式の H をパリティ検査行列にもつハミング符号 C を考える。 C における符号語 $\mathbf{w}$ の第 2, 4, 7 桁に誤りが生じたときの受信語を $\mathbf{w}'$ とする。このとき、

$$\mathbf{w}' H^t = \mathbf{h}_2 + \mathbf{h}_4 + \mathbf{h}_7 = (0, 1, 0) + (1, 0, 1) + (1, 1, 1) = \mathbf{0}$$

となる。よって $\mathbf{w}'$ は C の符号語となり 3 重誤り検出はできない。

演習問題 5.4

- (i) 例 5.9(3) でみたように C_1 は線形符号である。また、 $d_H(C) = 2$ であるのでハミング符号ではない。(もし C がハミング符号であるとする、例題 5.3 の (2) よりハミング符号は 2 重誤り検出可能であり、定理 5.3 より $d_H(C) \geq 3$ となり矛盾する。)
- (ii) C_2 は和で閉じているので線形符号である。基底として (例えば) $\langle 0011101, 0101011, 1000111 \rangle$ がとれ、これより生成行列 G は

$$G = \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

となる。補題 5.1 より $\dim(C_2) = \text{rank}(G) = 3$ であり、 $|C_2| = 2^3 - 1$ であるか

ら, C_2 はハミング符号でもある.

2. 定義により, 任意の要素 w, w' に対して,

$$\begin{aligned} d_H(w, w') &= w \text{ と } w' \text{ において値の異なる桁の個数} \\ &= (w - w') \text{ においてゼロでない桁の個数} \\ &= \text{weight}(w - w') \end{aligned}$$

である.

演習問題 5.5

1. $\text{rank}(H)=3$ であるから H は (6×3) 線形符号 C であり, $C = \{w \in F_2^6 \mid wH^t = \mathbf{0}\}$ より

$$C = \{000000, 111100, 101010, 010110, 011001, 100101, 110011, 001111\}$$

をえる. また生成行列は

$$G = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix}$$

となる.

2. G の行ベクトルから C の基底は $\langle w_1, w_2, w_3 \rangle$, ここで $w_1 = 11001, w_2 = 01010, w_3 = 01110$. 双対符号 C_{dual} の次元は $2(= 5-3)$ であり基底を $\langle u_1, u_2 \rangle$ とする. ここですべての $i = 1, 2, 3$ に対して $(u, w_i) = 0$ となる一次独立な 2 つのベクトルを計算し, $u_1 = 01011, u_2 = 11010$ を選ぶとする. このとき「 C_{dual} の生成行列 $G_{dual} = C$ のパリティ検査行列の転置行列」であることに留意すると求めたいパリティ検査行列 H は

$$H = G_{dual}^t = \begin{bmatrix} 1 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{bmatrix}^t$$

となる. また, $C_{dual} = \{uG_{dual} \mid u \in F_2^2\} = \{00000, 01011, 10001, 11010\}$.

演習問題 5.6

1. C のパリティ検査行列は

$$H = \begin{bmatrix} 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 1 \end{bmatrix}$$

2. H のすべての列ベクトルが異なることから C の最小距離は 3 以上であり, 第 1 列 + 第 3 列 + 第 4 列 = $\mathbf{0}$ であるから最小距離は 3 である. よって, C は 1

誤り語 e	シンδροーム s
000000	000
100000	111
010000	101
001000	100
000100	011
000010	010
000001	001

重誤り訂正可能であり，2重誤り検出可能である．

3. 誤り語とシンδροームの対応表は以下のようになる． $y = 111101$ に対するシンδροームは

$$s = (111101)H^t = (100)$$

よって誤り語は 00100 であり，符号語は 110101 (= 111101 - 001000)．またシンδροームの値がないのは 110 であるから， $xH^t = (110)$ を満たす x は 111111 である．

演習問題 5.7

1. $(x^n - 1)$ を $G(x)$ で割ったときの商を $s(x)$ ，余りを $r(x)$ とすると

$$(x^n - 1) - r(x) = s(x)G(x) \quad (\text{ここで } \deg r(x) < \deg G(x))$$

である．両辺を $(x^n - 1)$ で割った余りを考えると，“ $r(x)$ ” は $s(x)G(x)$ を $(x^n - 1)$ で割った剰余式である，ことが分かる．ここで定理 5.6 の証明において示した命題：『任意の多項式 $s(x)$ に対して， $s(x)G(x)$ を $(x^n - 1)$ で割った“剰余式”は符号多項式である』を用いると， $r(x)$ は符号多項式である．このとき $\deg r(x) < \deg G(x)$ かつ $G(x)$ はゼロでない最小次数の符号多項式であるから $r(x) = 0$ となり，結局 $(x^n - 1)$ は $G(x)$ で割り切れる．

2. 生成多項式を与える符号は 1100 であるから $G(x) = 1 + x$ となる．ここで $k = 4 - \deg G = 3$ であるから，基底となる符号多項式は

$$1 + x, x(1 + x) = x + x^2, x^2(1 + x) = x^2 + x^3$$

の 3 つである．よって基底は $\langle 1100, 0110, 0011 \rangle$ となる．

演習問題 5.8

1. まず $m = \deg G = 3$ ，よって $k = 4 (= 7 - 3)$ であることに留意せよ．

$p(\mathbf{y}) = p(1010) = 1 + x^2$ であるから, $x^3 p(\mathbf{y}) = x^3 + x^5$ となる. これを $G(x)$ で割った余りは $r(x) = x^2$ であり, これを多項式とする検査ビットは $\mathbf{x} = 001$ となる. よって送信語は

$$\mathbf{w} = \mathbf{x}\mathbf{y} = 0011010.$$

2. $H(x) = (x^7 - 1)/G(x) = 1 + x + x^2 + x^4$.

3. 命題 5.5 より, 与えられた多項式 $h(x)$ に対して $h(x)H(x)$ が $(x^7 - 1)$ で割り切れるならば, $h(x)$ は C の符号多項式である. (i) $p(\mathbf{z}_1)H(x)$ は $(x^7 - 1)$ で割り切れるので誤りなし. (ii) $p(\mathbf{z}_2)H(x)$ は $(x^7 - 1)$ で割り切れないので誤りあり.

演習問題 5.9

α^i を根にもつ $GF(2)$ 上の任意の関数を $f(x) = a_0 + a_1x + \cdots + a_nx^n$ とすると $f(\alpha^i) = 0$ である. このとき $a \in GF(2)$ に関して $a + a = 0, a^2 = a$ であることから

$$(f(x))^2 = (a_0 + a_1x + \cdots + a_nx^n)^2 = a_0 + a_1x^2 + \cdots + a_n(x^2)^n = f(x^2)$$

これより $f(\alpha^{2i}) = (f(\alpha^i))^2 = 0$. すなわち α^{2i} も $f(x)$ の根であるから, 一般に α^{2^j} ($j \geq 0$) も根となる. よって, これらはすべて α^i の最小多項式 $m_i(x)$ の根である. 一方, d_i の取り方 (最小性) と最小多項式の定義より $m_i(x)$ は $\alpha^i, \alpha^{2i}, \alpha^{2^2i}, \dots, \alpha^{i2^{d_i-1}}$ のすべてを, かつこれだけを根とする多項式となる.

演習問題 5.10

問題の BCH 符号のパリティ検査行列 H は (5.42) 式で与えられる. H の任意の $2t$ 個の列からなる行列の行列式を

$$D = \begin{vmatrix} \alpha^{i_1} & \alpha^{i_2} & \alpha^{i_3} & \cdots & \alpha^{i_{2t}} \\ (\alpha^2)^{i_1} & (\alpha^2)^{i_2} & (\alpha^2)^{i_3} & \cdots & (\alpha^2)^{i_{2t}} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (\alpha^{2t})^{i_1} & (\alpha^{2t})^{i_2} & (\alpha^{2t})^{i_3} & \cdots & (\alpha^{2t})^{i_{2t}} \end{vmatrix} \quad (49)$$

を考え, これを計算すると

$$D = \alpha^{i_1+i_2+\cdots+i_{2t}} \prod_{1 \leq p < q \leq 2t} (\alpha^p - \alpha^q)$$

である. ここで $\alpha^{i_1}, \alpha^{i_2}, \dots, \alpha^{i_{2t}}$ はすべて異なるので $D \neq 0$, すなわち D は正則となる. これより H から任意に選んだ $2t$ 個の列ベクトルが一次独立であることが分かる.

さて, 符号語 $\mathbf{w} = c_0c_1 \cdots c_{n-1}$ は $\mathbf{w}H^t = \mathbf{0}$ を満たすので,

$$c_0\mathbf{h}_0^t + c_1\mathbf{h}_1^t + \cdots + c_{n-1}\mathbf{h}_{n-1}^t = \mathbf{0}$$

ここで $H = [h_0 \ h_1 \ \cdots \ h_{n-1}]$ である．ここで w が最小重みを与えるとする．
 上述のように H の列は任意の $2t$ 個が一次独立であるので， $c_0, \dots, c_{n-1}$ のうちで 0 でないものが少なくとも $2t+1$ 個存在する．よって最小重み（すなわち最小ハミング距離）は $(2t+1)$ 以上となる．

演習問題 5.11

1. 与えられた各 c_i から $d = 2891$ と $n = 10807$ を用いて $m_i \equiv c_i^d \pmod{n}$ により復号化すると，

$$\begin{aligned}\overline{m} &= m_1 m_2 m_3 m_4 m_5 = 2515\ 2100\ 0715\ 2000\ 0920 \\ &= 25\ 15\ 21\ 00\ 07\ 15\ 20\ 00\ 09\ 20\end{aligned}$$

これより $m = \text{YOU GOT IT}$ となる．

2. A はまず自己の復号化鍵 $A_d = 2891$ を用いて m から $D_{A_d}(m) = 2109^{2891} \equiv 5482 \pmod{10807}$ と計算する．さらに B の公開鍵 $B_k = (9797, 1543)$ を用いると暗号文は $c = E_{B_k}(D_{A_d}(m)) = 5482^{1543} \equiv 7017 \pmod{9797}$ となる．

$c = 7017$ を受け取った B は自己の復号化鍵 $B_d = 1207$ により $D_{B_d}(c) = 7017^{1207} \equiv 5482 \pmod{9797}$ を得て，あとは公開されている A の暗号化鍵 $A_k = 11$ により $\overline{m} = 5482^{11} \equiv 2109 \pmod{10807}$ と（すなわち平文 “UI” を）復号化する．