

「離散数学・オートマトン」演習問題 06 (解答例)

2020/11/17

1 グラフ

課題 1 以下のグラフ $G = (V, E)$ を図示しなさい。

$$V = \{v_0, v_1, v_2, v_3, v_4, v_5\}$$

$$E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$$

$$\partial^+ e_0 = v_0,$$

$$\partial^- e_0 = v_1$$

$$\partial^+ e_1 = v_0,$$

$$\partial^- e_1 = v_4$$

$$\partial^+ e_2 = v_1,$$

$$\partial^- e_2 = v_5$$

$$\partial^+ e_3 = v_1,$$

$$\partial^- e_3 = v_3$$

$$\partial^+ e_4 = v_2,$$

$$\partial^- e_4 = v_0$$

$$\partial^+ e_5 = v_2,$$

$$\partial^- e_5 = v_1$$

$$\partial^+ e_6 = v_2,$$

$$\partial^- e_6 = v_4$$

$$\partial^+ e_7 = v_3,$$

$$\partial^- e_7 = v_2$$

$$\partial^+ e_8 = v_4,$$

$$\partial^- e_8 = v_5$$

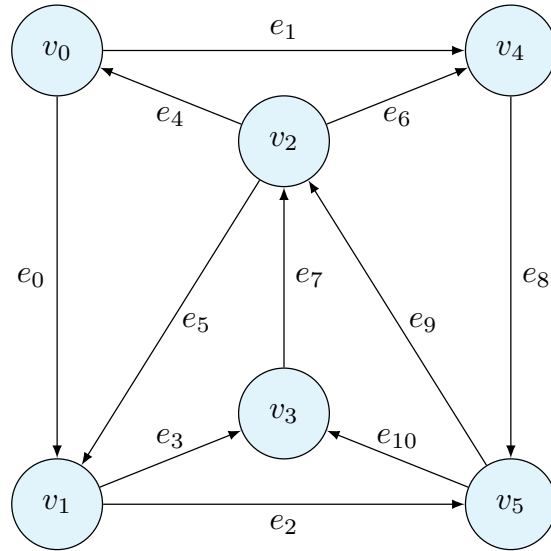
$$\partial^+ e_9 = v_5,$$

$$\partial^- e_9 = v_3$$

$$\partial^+ e_{10} = v_5,$$

$$\partial^- e_{10} = v_3$$

解答例



Python のライブラリ `nexworkx` を使うことで、グラフを作図することができる。以下の Github から取得できる。

<https://github.com/discrete-math-saga/Graph>

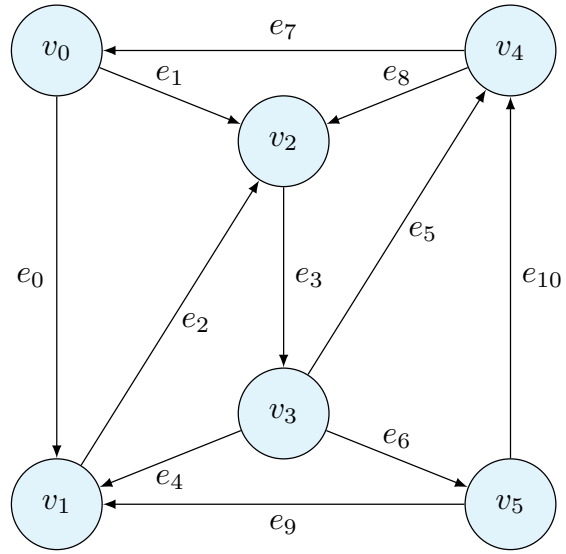
課題 2 以下のグラフ $G = (V, E)$ を図示しなさい。

$$V = \{v_0, v_1, v_2, v_3, v_4, v_5\}$$

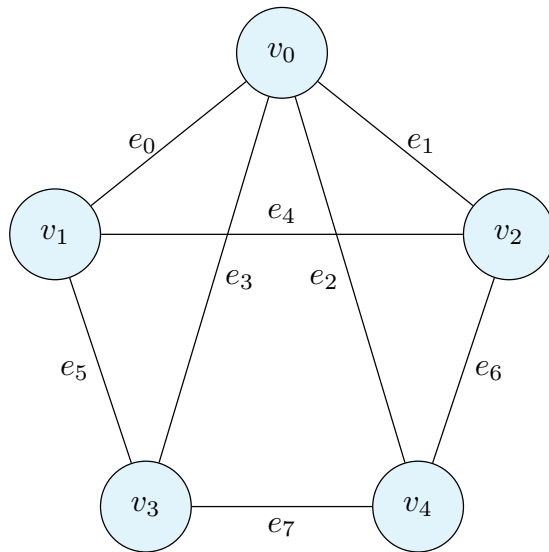
$$E = \{e_0, e_1, e_2, e_3, e_4, e_5, e_6, e_7, e_8, e_9, e_{10}\}$$

$\partial^+ e_0 = v_0,$	$\partial^- e_0 = v_1$	$\partial^+ e_1 = v_0,$	$\partial^- e_1 = v_2$
$\partial^+ e_2 = v_1,$	$\partial^- e_2 = v_2$	$\partial^+ e_3 = v_2,$	$\partial^- e_3 = v_3$
$\partial^+ e_4 = v_3,$	$\partial^- e_4 = v_1$	$\partial^+ e_5 = v_3,$	$\partial^- e_5 = v_4$
$\partial^+ e_6 = v_3,$	$\partial^- e_6 = v_5$	$\partial^+ e_7 = v_4,$	$\partial^- e_7 = v_0$
$\partial^+ e_8 = v_4,$	$\partial^- e_8 = v_2$	$\partial^+ e_9 = v_5,$	$\partial^- e_9 = v_1$
$\partial^+ e_{10} = v_5,$	$\partial^- e_{10} = v_4$		

解答例



課題 3 以下のグラフに対して、Hamilton 閉路を列挙せよ。



解答例 以下の 8 種類の Hamilton 閉路がある。逆回りも含んでいる。

v_0, v_1, v_2, v_4, v_3
 v_0, v_1, v_3, v_4, v_2
 v_0, v_2, v_1, v_3, v_4
 v_0, v_2, v_4, v_3, v_1
 v_0, v_3, v_1, v_2, v_4
 v_0, v_3, v_4, v_2, v_1
 v_0, v_4, v_2, v_1, v_3
 v_0, v_4, v_3, v_1, v_2

閉路を列挙する Python プログラムを示す。これも、前問同様の Github から取得できる。

```
1 def enumerateHamilton(start,G):
2     VHamilton = list()
3     VHamilton.append(start)
4     circuits = list()
5     enumerateHamiltonSub(start,start,VHamilton,G,circuits)
6     return circuits
7
8 def enumerateHamiltonSub(currentNode,startNode,VHamilton,G, circuits):
9     for edge in nx.edges(G,currentNode):
10        (f,t) = edge
11        if (t is startNode) and (len(G.nodes) == len(VHamilton)):
12            circuits.append(VHamilton)
13        else:
14            if t not in VHamilton:
15                E = list(VHamilton)
16                E.append(t)
17                enumerateHamiltonSub(t,startNode,E,G,circuits)
```