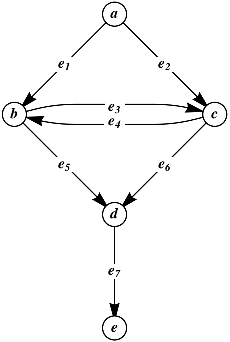
Directed Graphs

Definition. A **directed graph** (or **digraph**) is a graph $G = (V, E)$, where every edge $e = vw$ is directed from one vertex to another:

$$e : v \to w \qquad \text{or} \qquad e : w \to v.$$

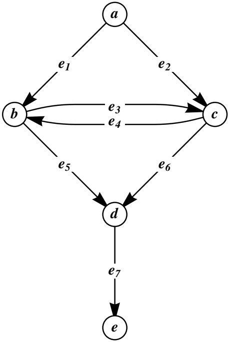


Directed Graphs

Definition. A **directed graph** (or **digraph**) is a graph $G = (V, E)$, where every edge $e = vw$ is directed from one vertex to another:

$$e : v \to w \quad \text{or} \quad e : w \to v.$$

Remark. An edge $e : v \to w$ is different from $e' : w \to v$ and a digraph including both is not considered to have multiple edges.



Directed Graphs

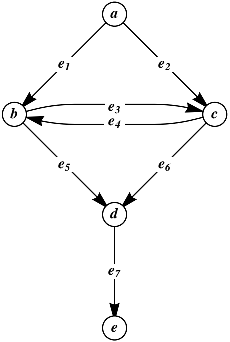
Definition. A **directed graph** (or **digraph**) is a graph $G = (V, E)$, where every edge $e = vw$ is directed from one vertex to another:

$$e : v \to w \qquad \text{or} \qquad e : w \to v.$$

Remark. An edge $e : v \to w$ is different from $e' : w \to v$ and a digraph including both is not considered to have multiple edges.

Definition. The **in-degree** of a vertex v is the number of edges directed *toward* v.

Definition. The **out-degree** of a vertex v is the number of edges directed *away from* v.



Directed Graphs

Definition. A **directed graph** (or **digraph**) is a graph $G = (V, E)$, where every edge $e = vw$ is directed from one vertex to another:

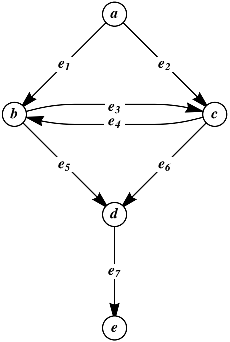
$$e : v \to w \qquad \text{or} \qquad e : w \to v.$$

Remark. An edge $e : v \to w$ is different from $e' : w \to v$ and a digraph including both is not considered to have multiple edges.

Definition. The **in-degree** of a vertex v is the number of edges directed *toward* v.

Definition. The **out-degree** of a vertex v is the number of edges directed *away from* v.

Important. Any **path** / **cycle** / **walk** in a digraph must respect the direction on every edge.



Directed Graphs

Definition. A **directed graph** (or **digraph**) is a graph $G = (V, E)$, where every edge $e = vw$ is directed from one vertex to another:

$$e : v \to w \quad \text{or} \quad e : w \to v.$$

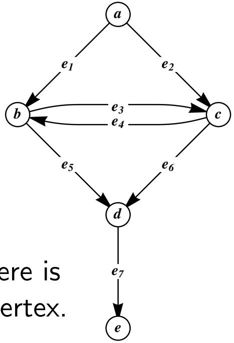
Remark. An edge $e : v \to w$ is different from $e' : w \to v$ and a digraph including both is not considered to have multiple edges.

Definition. The **in-degree** of a vertex v is the number of edges directed *toward* v.

Definition. The **out-degree** of a vertex v is the number of edges directed *away from* v.

Important. Any **path** / **cycle** / **walk** in a digraph must respect the direction on every edge.

Definition. A digraph is **strongly connected** if there is a directed path from every vertex to every other vertex.



Generalizations to directed graphs

Definition. A **directed pseudograph** allows loops and multiple edges.

Generalizations to directed graphs

Definition. A **directed pseudograph** allows loops and multiple edges.

We can generalize Theorems 3.1.1 and 3.1.2 to directed pseudographs:

Let G be a directed pseudograph that is strongly connected*.

G has an Eulerian circuit

$$\Updownarrow$$

the in-degree of every vertex equals
the out-degree of every vertex.

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession
$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession
$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition. A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession
$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition. A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Definition. A **de Bruijn sequence** of order n on $\mathcal{A}$ is word of length k^n in which every n-length word occurs as a consecutive subsequence.

Example. $S = 0000110101111001$ is a *binary* de Bruijn seq. of order 4.

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession

$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition. A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Definition. A **de Bruijn sequence** of order n on $\mathcal{A}$ is word of length k^n in which every n-length word occurs as a consecutive subsequence.

Example. $S = 0000110101111001$ is a *binary* de Bruijn seq. of order 4.

EVERY binary sequence of length 4 is present. (We allow cycling.)

0000	0010	0100	0110	1000	1010	1100	1110
0001	0011	0101	0111	1001	1011	1101	1111

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession

$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition. A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Definition. A **de Bruijn sequence** of order n on $\mathcal{A}$ is word of length k^n in which every n-length word occurs as a consecutive subsequence.

Example. $S = 0000110101111001$ is a *binary* de Bruijn seq. of order 4.

EVERY binary sequence of length 4 is present. (We allow cycling.)

0000	0010	0100	0110	1000	1010	1100	1110
0001	0011	0101	0111	1001	1011	1101	1111

This is the most compact way to represent these sixteen sequences.

Application: de Bruijn sequences

Definition. An **alphabet** is a set $\mathcal{A} = \{a_1, \ldots, a_k\}$.
Definition. A **sequence** or **word** from $\mathcal{A}$ is a succession
$S = s_1 s_2 s_3 \cdots s_l$, where each $s_i \in \mathcal{A}$; l is the **length** of S.

Definition. A sequence is called a **binary sequence** when $\mathcal{A} = \{0, 1\}$.

Definition. A **de Bruijn sequence** of order n on $\mathcal{A}$ is word of length k^n in which every n-length word occurs as a consecutive subsequence.

Example. $S = 0000110101111001$ is a *binary* de Bruijn seq. of order 4.

EVERY binary sequence of length 4 is present. (We allow cycling.)

0000	0010	0100	0110	1000	1010	1100	1110
0001	0011	0101	0111	1001	1011	1101	1111

This is the most compact way to represent these sixteen sequences.

Theorem. A de Bruijn sequence of order n on $\mathcal{A}$ always exists.

de Bruijn graphs

Definition. The **de Bruijn graph** of order n
on the alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph.

de Bruijn graphs

Definition. The **de Bruijn graph** of order n
on the alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph.
Its vertices are labeled by words of $\mathcal{A}$ of length $n - 1$.

de Bruijn graphs

Definition. The **de Bruijn graph** of order n on the alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph.

Its vertices are labeled by words of $\mathcal{A}$ of length $n-1$.

Each vertex has k out-edges labeled by the letters of $\mathcal{A}$:

$$\boxed{b_1 b_2 \cdots b_{n-1}} \xrightarrow{a_i} \boxed{b_2 \cdots b_{n-1} a_i}$$

(Remove the first letter and append a_i at the end.)

de Bruijn graphs

Definition. The **de Bruijn graph** of order n on the alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph.

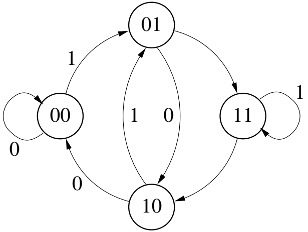
Its vertices are labeled by words of $\mathcal{A}$ of length $n-1$.

Each vertex has k out-edges labeled by the letters of $\mathcal{A}$:

$$\boxed{b_1 b_2 \cdots b_{n-1}} \xrightarrow{a_i} \boxed{b_2 \cdots b_{n-1} a_i}$$

(Remove the first letter and append a_i at the end.)

Examples. The binary de Bruijn graph of order 3



de Bruijn graphs

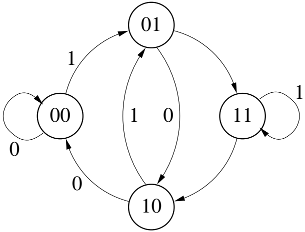
Definition. The **de Bruijn graph** of order n
on the alphabet $\mathcal{A} = \{a_1, a_2, \ldots, a_k\}$ is a directed pseudograph.
Its vertices are labeled by words of $\mathcal{A}$ of length $n - 1$.
Each vertex has k out-edges labeled by the letters of $\mathcal{A}$:

$$\boxed{b_1 b_2 \cdots b_{n-1}} \xrightarrow{a_i} \boxed{b_2 \cdots b_{n-1} a_i}$$

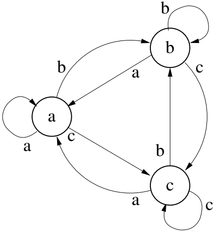
(Remove the first letter and append a_i at the end.)

Examples.

The binary de Bruijn graph of order 3



The de Bruijn graph of order 2 on the alphabet $\mathcal{A} = \{a, b, c\}$.



Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C.

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) $=$ out-degree(v) for all $v \in V$. (Why?)

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) $=$ out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) = out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

- S is of length k^n.

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) = out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

- S is of length k^n.
- Every n-length word occurs as a consecutive subsequence of S.

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) = out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

- S is of length k^n.
- Every n-length word occurs as a consecutive subsequence of S.

By construction, the sequence of the $n-1$ labels of edges visited before arriving at a vertex is **exactly** the label of the vertex.

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) $=$ out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

- S is of length k^n.
- Every n-length word occurs as a consecutive subsequence of S.

By construction, the sequence of the $n-1$ labels of edges visited before arriving at a vertex is **exactly** the label of the vertex.

Recording the label of an edge e in C completes a word of length n:

(label of origin vertex) $+$ (label of edge)

Proof that a de Bruijn sequence always exists

Claim. The de Bruijn graph G of order n on $\mathcal{A}$ has an Eulerian circuit C. This follows because G is strongly connected. (Why?)

AND in-degree(v) = out-degree(v) for all $v \in V$. (Why?)

Construct a sequence S: Follow C and record the edge labels.

Claim. S is a de Bruijn sequence of order n on $\mathcal{A}$.

- S is of length k^n.
- Every n-length word occurs as a consecutive subsequence of S.

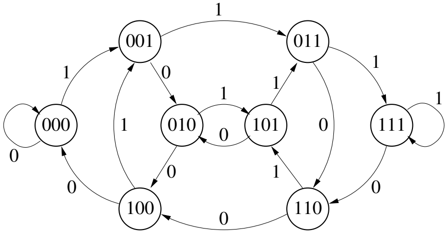
By construction, the sequence of the $n-1$ labels of edges visited before arriving at a vertex is **exactly** the label of the vertex.

Recording the label of an edge e in C completes a word of length n:

(label of origin vertex) + (label of edge)

This is a different word for every edge! So every word appears in S.

Example: The binary de Bruijn graph of order 4



1. Find an Eulerian circuit in this graph.
2. Write down the corresponding sequence.
3. Verify that it is a de Bruijn sequence. (use chart, p.63)
4. Convince yourself that the name of a vertex is the same as the sequence formed by the three previous edges.