Comments on Notation

Sometimes, instead of numbering equations, key statements etc., we have marked them with symbols such as $(*), (**), (\surd)$. These marks are used over and over again and have validity only within a local area such as a paragraph, a proof or the solution to a problem.

In some cases, where there is no room for confusion, the same symbol denotes different objects. For instance, usually B denotes a bipartite graph. But in Chapter **??**, B denotes a base of a matroid- elsewhere a base is always denoted by b. The symbol E is used for the edge set of a graph, in particular a bipartite graph. But $E(X), X \subseteq V(\mathcal{G})$ denotes the set of edges with both endpoints within X, while $E_L(X), X \subseteq V_L$, in the case of a bipartite graph, denotes the set of all vertices adjacent only to vertices in X.

We have often used brackets to write two statements in one.
Example: We say that set X is **contained** in Y (**properly contained in** Y), if every element of X is also a member of Y (every element of X is a member of Y and $X \neq Y$) and denote it by $X \subseteq Y (X \subset Y)$. This is to be read as the following two statements.

i. We say that set X is **contained** in Y, if every element of X is also a member of Y and denote it by $X \subseteq Y$.

ii. We say that set X is **properly contained in** Y, if every element of X is a member of Y and $X \neq Y$ and denote it by $X \subset Y$.

List of Commonly Used Symbols

Sets, Partitions, Partial Orders

$\{e_1, e_2, \ldots, e_n\}$	*set whose elements are* $e_1, e_2, \ldots, e_n$
$\{x_i : i \in I\}$	*set whose members are* $x_i,\ i \in I$
$(x_i : i \in I)$	*a family (used only in Chapters 2 and 11)*
$x \in X$	*element* x *belongs to set* X
$x \notin X$	*element* x *does not belong to set* X
$\forall x$ *or* $\forall\ x$	*for all elements* x
$\exists x$	*there exists an element* x
$X \subseteq Y$	*set* X *is contained in set* Y
$X \subset Y$	*set* X *is properly contained in set* Y
$X \cup Y$	*union of sets* X *and* Y
$X \cap Y$	*intersection of sets* X *and* Y
$X \uplus Y$	*disjoint union of sets* X *and* Y
$\bigcup_{i=1}^{n} X_i$	*union of the sets* X_i
$\biguplus_{i=1}^{n} X_i$	*disjoint union of the sets* X_i
$X - Y$	*set of elements in* X *but not in* Y
$\bar{X}$	*complement of* X
$X \times Y$	*cartesian product of sets* X *and* Y
$X \oplus Y$	*direct sum of sets* X *and* Y
2^S	*collection of subsets of* S
$\mid X \mid$	*size of the subset* X
$(P, \preceq)$	*preorder on* P
$(P, \leq)$	*partial order on* P
Π	*partition* Π
Π_N	*partition that has* N *as a block and all blocks except* N *as singletons*
$\mathcal{P}_X$	*collection of all partitions of* X
$\Pi \leq \Pi'$	*partition* Π *is finer than* Π'

$\Pi \vee \Pi'$	*finest partition coarser than both* Π *and* Π'
$\Pi \wedge \Pi'$	*coarsest partition finer than both* Π *and* Π'

Functions,Set Functions and Operations on Functions

$f(\cdot)$	*function* $f(\cdot)$
$f/Z(\cdot), f(\cdot)$ *on* S	*restriction of* $f(\cdot)$ *to* $Z \subseteq S$
$gf(X), g \circ f(X)$	$g(f(X))$
$(f_1 \oplus f_2)(\cdot)$	*direct sum of functions* $f_1(\cdot)$ *and* $f_2(\cdot)$
$f_{fus \cdot \Pi}(\cdot), f(\cdot)$ *on* 2^S,	*fusion of* $f(\cdot)$ *relative to* Π
	i.e., $f_{fus.\Pi}(X_f)$
	$\equiv f(\bigcup_{T \in X_f} T), X_f \subseteq \Pi$
$f/\mathbf{X}(\cdot), f(\cdot)$ *on* 2^S	*restriction of* $f(\cdot)$ *to* $2^X, X \subseteq S$
	(usually called) restriction of $f(\cdot)$ *to* X
$f \diamond \mathbf{X}(\cdot), f(\cdot)$ *on* 2^S	*contraction of* $f(\cdot)$ *to* X
	$f \diamond \mathbf{X}(Y) \equiv f((S-X) \cup Y) - f(S-X)$
$f^d(\cdot), f(\cdot)$ *on* 2^S	*contramodular dual of* $f(\cdot)$
	$f^d(X) \equiv f(S) - f(S-X)$
$f^*(\cdot), f(\cdot)$ *on* 2^S	*comodular dual of* $f(\cdot)$
	(with respect to weight function $\alpha(\cdot)$*)*
	$f^*(X) \equiv \alpha(X) - (f(S) - f(S-X))$
$P_f, f(\cdot)$ *on* 2^S	*polyhedron associated with* $f(\cdot)$
	$\mathbf{x} \in P_f$ *iff* $x(X) \leq f(X) \quad \forall X \subseteq S$
$P^d_f, f(\cdot)$ *on* 2^S	*dual polyhedron associated with* $f(\cdot)$
	$\mathbf{x} \in P^d_f$ *iff* $x(X) \geq f(X) \quad \forall X \subseteq S$

Vectors and Matrices

$\mathcal{F}, \Re, \mathcal{C}, \Re_+$	*field* $\mathcal{F}$, *real field, complex field,*
	set of nonnegative reals
$\sum x_i$	*summation of elements* x_i
$\mathbf{f}$	*vector* $\mathbf{f}$
$\mathcal{V}$	*vector space* $\mathcal{V}$
$\mathcal{V}^\perp$	*vector space complementary orthogonal to* $\mathcal{V}$
$\mathbf{x}_1 \oplus \mathbf{x}_2$	*direct sum of* $\mathbf{x}_1$ *and* $\mathbf{x}_2$ *(vector obtained by*

	adjoining components of vectors $\mathbf{x}_1$ *and* $\mathbf{x}_2$)
$\mathcal{V}_S \oplus \mathcal{V}_T, S \cap T = \emptyset$	*direct sum of* $\mathcal{V}_S$ *and* $\mathcal{V}_T$ (*obtained by collecting all possible direct sums of vectors in* $\mathcal{V}_S$ *and* $\mathcal{V}_T$)

$dim(\mathcal{V}), r(\mathcal{V})$	*dimension of vector space* $\mathcal{V}$
$d(\mathcal{V}, \mathcal{V}')$	$r(\mathcal{V}+\mathcal{V}') - r(\mathcal{V}\cap\mathcal{V}')$
$\mathbf{A}(i,j)$	i, j^{th} *entry of matrix* $\mathbf{A}$
$\mathbf{A}^T$	*transpose of matrix* $\mathbf{A}$
$\mathbf{A}^{-1}$	*inverse of matrix* $\mathbf{A}$
$< \mathbf{f}, \mathbf{g} >$	*dot product of vectors* $\mathbf{f}, \mathbf{g}$
$\mathcal{R}(\mathbf{A})$	*row space of* $\mathbf{A}$
$\mathcal{C}(\mathbf{A})$	*column space of* $\mathbf{A}$
$det(\mathbf{A})$	*determinant of* $\mathbf{A}$

Graphs and Vector Spaces

$\mathcal{G}$	*graph* $\mathcal{G}$
$V(\mathcal{G})$	*vertex set of* $\mathcal{G}$
$E(\mathcal{G})$	*edge set of* $\mathcal{G}$
t	*a tree*
f	*a forest*
$\bar{t}$	*cotree* $(E(\mathcal{G}) - t)$ *of* $\mathcal{G}$
$\bar{f}$	*coforest* $(E(\mathcal{G}) - f)$ *of* $\mathcal{G}$
$L(e, f)$	$f-$*circuit of* e *with respect to* f
$B(e, f)$	$f-$*cutset of* e *with respect to* f
$r(\mathcal{G})$	*rank of* $\mathcal{G}$ (= *number of edges in a forest of* $\mathcal{G}$)
$\nu(\mathcal{G})$	*nullity of* $\mathcal{G}$ (= *number of edges in a coforest of* $\mathcal{G}$)
$\mathcal{G}openT$	*graph obtained from* $\mathcal{G}$ *by opening and removing edges* T
$\mathcal{G}shortT$	*graph obtained from* $\mathcal{G}$ *by shorting and removing edges* T
$\mathcal{G}\cdot T$	*graph obtained from* $\mathcal{G}open(E(\mathcal{G}) - T)$ *by removing isolated vertices,*

	restriction of $\mathcal{G}$ *to* T
$\mathcal{G} \times T$	*graph obtained from* $\mathcal{G}short(E(\mathcal{G}) - T)$ *by removing isolated vertices,*
	contraction of $\mathcal{G}$ *to* T

$\mathcal{G}_1 \cong \mathcal{G}_2$	$\mathcal{G}_1$ *is* $2-$*isomorphic to* $\mathcal{G}_2$
$r(T)$	$r(\mathcal{G} \cdot T)$
$\nu(T)$	$\nu(\mathcal{G} \times T)$
$\mathcal{H}$	*hypergraph* $\mathcal{H}$
$B(V_L, V_R, E)$	*bipartite graph with left vertex set* V_L, *right vertex set* V_R *and edge set* E
$\mathbf{A}$	*(usually) incidence matrix*
$\mathbf{A_r}$	*reduced incidence matrix*
$\mathbf{Q}_f$	*fundamental cutset matrix of forest* f
$\mathbf{B}_f$	*fundamental circuit matrix of forest* f
KCE	*Kirchhoff's current equations*
KCL	*Kirchhoff's current Law*
KVE	*Kirchhoff's voltage equations*
KVL	*Kirchhoff's voltage Law*
$\mathcal{V}_i(\mathcal{G})$	*solution space of* KCE *of* $\mathcal{G}$
$\mathcal{V}_v(\mathcal{G})$	*solution space of* KVE *of* $\mathcal{G}$
$\mathcal{V} \cdot T$	*restriction of vector space* $\mathcal{V}$ *to* T
$\mathcal{V} \times T$	*contraction of vector space* $\mathcal{V}$ *to* T
$\xi(T)$ *for* $\mathcal{V}$	$r(\mathcal{V} \cdot T) - r(\mathcal{V} \times T)$

Flow Graphs

$F(\mathcal{G}) \equiv (\mathcal{G}, \mathbf{c}, s, t)$	*flow graph on graph* $\mathcal{G}$ *with capacity function* $\mathbf{c}$, *source* s, *sink* t
(A, B)	*cut*(A, B)
$c(A, B)$	*capacity of cut*(A, B)
$f(A, B)$	*flow across cut*(A, B), *from* A *to* B
$\|\mathbf{f}\|$	*value of flow* $\mathbf{f}$
$F(B, \mathbf{c}_L, \mathbf{c}_R)$	*flowgraph associated with bipartite graph* B *with source to left vertex capacity* $\mathbf{c}_L$, *right*

vertex to sink capacity $\mathbf{c}_R$
and (left to right) bipartite graph edge capacity ∞

Matroids

$\mathcal{M} \equiv (S, \mathcal{I})$	*matroid* $\mathcal{M}$
$\mathcal{I}$	*collection of independent sets*
$\mathcal{M}^*$	*dual of the matroid* $\mathcal{M}$
B	*(only in Chapter 4) base of a matroid*
$L(e, B)$	$f-$*circuit of* e *with respect to base* B
$B(e, B)$	$f-$*bond of* e *with respect to base* B
$r(T)$	*rank of the subset* T *in the given matroid*
$r(\mathcal{M})$	*rank of the underlying set of the matroid*
$\nu(T)$	*rank of the subset* T *in the dual of the given matroid*
$\nu(\mathcal{M})$	*rank of the underlying set in the dual matroid*
$\mathcal{M}(\mathcal{G})$	*polygon matroid of the graph* $\mathcal{G}$ *(bases are* $forests$*)*
$\mathcal{M}^*(\mathcal{G})$	*bond matroid of the graph* $\mathcal{G}$ *(bases are* $coforests$*)*
$\mathcal{M}(\mathcal{V})$	*matroid whose bases are maximal independent columns of a representative matrix of* $\mathcal{V}$
$\mathcal{M}^*(\mathcal{V})$	*dual of* $\mathcal{M}(\mathcal{V})$
$\int(X)$	*span (closure) of the subset* X *in the matroid*
$\mathcal{M} \cdot T$	*restriction of* $\mathcal{M}$ *to* T
$\mathcal{M} \times T$	*contraction of* $\mathcal{M}$ *to* T
$\mathcal{M}_1 \vee \mathcal{M}_2$	*union of matroids* $\mathcal{M}_1$ *and* $\mathcal{M}_2$

Electrical Networks

$\mathbf{v}$	*voltage vector*
$\mathbf{i}$	*current vector*
$\mathcal{N}$	*electrical network*
$\mathcal{N}_{AP}$	*electrical multiport with port set* P *and remaining edge set* A
$\mathcal{E}$	*set of voltage sources in the network*

$\mathcal{J}$	*set of current sources in the network*
R	*resistance, also collection of resistors or 'current controlled voltage' elements in the network*

G	*conductance, also collection of 'voltage controlled current' elements in the network*
L	*inductance, also collection of inductors in the network*
$\mathcal{L}$	*mutual inductance matrix*
C	*capacitance, also collection of capacitors in the network*
vcvs	*voltage controlled voltage source*
vccs	*voltage controlled current source*
ccvs	*current controlled voltage source*
cccs	*current controlled current source*
$\mathcal{D}$	*device characteristic*
$\mathcal{D}_{AB}$	$(\mathbf{v}/A, \mathbf{i}/B),$ *where* $\mathbf{v}, \mathbf{i} \in \mathcal{D}$
$\mathcal{D}_A$	$\mathcal{D}_{AA}$
$\mathcal{D}_{AB} \times \mathcal{D}_{PQ}$	$\{(\mathbf{v}, \mathbf{i}),\ \mathbf{v} = \mathbf{v}_A \oplus \mathbf{v}_P, \mathbf{i} = \mathbf{i}_B \oplus \mathbf{i}_Q$ *where* $(\mathbf{v}_A, \mathbf{i}_B) \in \mathcal{D}_{AB}, (\mathbf{v}_P, \mathbf{i}_Q) \in \mathcal{D}_{PQ}\}$
δ_{AB}	$\{(\mathbf{v}_A, \mathbf{i}_B),\ \mathbf{v}_A$ *is any vector on* $A,\ \mathbf{i}_B$ *is any vector on* $B\}$

Implicit Duality

$\mathcal{K}_{SP} \leftrightarrow \mathcal{K}_P$	$\{f_S : f_S = f_{SP}/S,\ f_{SP} \in \mathcal{K}_{SP}\ \ s.t.\ f_{SP}/P \in \mathcal{K}_P\}$
$\mathcal{K}_{S_1} \leftrightarrow \mathcal{K}_{S_2}$	$\{\mathbf{f} : \mathbf{f} = \mathbf{f}_1/(S_1 - S_2) \oplus \mathbf{f}_2/(S_2 - S_1),\ \mathbf{f}_1 \in \mathcal{K}_{S_1}, \mathbf{f}_2 \in \mathcal{K}_{S_2}$ *and* $\mathbf{f}_1/S_1 \cap S_2 = \mathbf{f}_2/S_1 \cap S_2\}$
$\mathcal{K}_{S_1} \rightleftharpoons \mathcal{K}_{S_2}$	$\{\mathbf{f} : \mathbf{f} = \mathbf{f}_1/(S_1 - S_2) \oplus \mathbf{f}_2/(S_2 - S_1),\ \mathbf{f}_1 \in \mathcal{K}_{S_1}, \mathbf{f}_2 \in \mathcal{K}_{S_2}$ *and* $\mathbf{f}_1/S_1 \cap S_2 = -\mathbf{f}_2/S_1 \cap S_2\}$
$< \cdot, \cdot >$	*a* $q-$*bilinear operation, usually dot product*
$\mathcal{K}^*$	*collection of vectors* $q-$*orthogonal to those in* $\mathcal{K}$
$d(\mathcal{V}, \mathcal{V}')$	$r(\mathcal{V} + \mathcal{V}') - r(\mathcal{V} \cap \mathcal{V}')$
$\mathcal{K}^p$	*the collection of vectors polar to those in* $\mathcal{K}$

$\mathcal{K}^d$	*(only in Chapter 7) the collection of vectors integrally dual to those in* $\mathcal{K}$

Multiport Decomposition

$(\mathcal{V}_{E_1P_1},\cdots,\mathcal{V}_{E_kP_k};\mathcal{V}_P)$	$k-$ *multiport decomposition of* $\mathcal{V}_E$ (*i.e.*, $(\bigoplus_i \mathcal{V}_{E_iP_i}) \leftrightarrow \mathcal{V}_P = \mathcal{V}_E$)
$((\mathcal{V}_{E_jP_j})_k;\mathcal{V}_P)$	$(\mathcal{V}_{E_1P_1},\cdots,\mathcal{V}_{E_kP_k};\mathcal{V}_P)$
$((\mathcal{V}_{E_jP_j})_{j\in I};\mathcal{V}_{P_I})$	$(\cdots\mathcal{V}_{E_jP_j}\cdots;\mathcal{V}_{P_I})$ *where* $j \in I \subseteq \{1,\cdots,k\}$ *and* $P_I = \cup_{j\in I}P_j$
$(\mathcal{V}_{EP},P)$	*vector space on* $E \uplus P$ *with ports* P
$(\mathcal{V}_{E_1Q_1},\cdots,\mathcal{V}_{E_kQ_k};\mathcal{V}_{QP})$	*matched or skewed decomposition of* $(\mathcal{V}_{EP},P)$

Functions Associated with Graphs and Bipartite Graphs

$V(X), X \subseteq E(\mathcal{G})$	*set of endpoints of edges* X *in graph* $\mathcal{G}$
$\Gamma(X), X \subseteq V(\mathcal{G})$	*set of vertices adjacent to vertices in vertex subset* X *in graph* $\mathcal{G}$
$\Gamma_L(X), X \subseteq V_L$	*in* $B \equiv (V_L, V_R, E)$, *set of vertices adjacent to vertices in* X
$\Gamma_R(X), X \subseteq V_R$	*in* $B \equiv (V_L, V_R, E)$, *set of vertices adjacent to vertices in* X
$E(X), X \subseteq V(\mathcal{G})$	*set of edges with both endpoints in vertex subset* X *in graph* $\mathcal{G}$
$E_L(X), X \subseteq V_L$	*in* $B \equiv (V_L, V_R, E)$ *set of vertices adjacent* **only** *to vertices in* X
$E_R(X), X \subseteq V_R$	*in* $B \equiv (V_L, V_R, E)$ *set of vertices adjacent* **only** *to vertices in* X
$I(X), X \subseteq V(\mathcal{G})$	*set of edges with atleast one endpoint in vertex subset* X *in graph* $\mathcal{G}$
$cut(X), X \subseteq V(\mathcal{G})$	*set of edges with exactly one endpoint in vertex subset* X *in graph* $\mathcal{G}$
$w(\cdot)$	*usually a weight function*
$w_L(\cdot), w_R(\cdot)$	*weight functions on the left vertex set*

and on the right vertex set respectively
of a bipartite graph

Convolution and PP

$f * g(X)$	*convolution of* $f(\cdot)$ *and* $g(\cdot)$ $(min_{Y\subseteq X}[f(Y)+g(X-Y)])$
$\mathcal{B}_{\lambda_{f,g}}, f(\cdot), g(\cdot)$ *on* 2^S	*collection of sets which minimize* $\lambda f(X)+g(S-X)$ *over subsets of* S
$\mathcal{B}_\lambda$	$\mathcal{B}_{\lambda_{f,g}}$
X^λ, X_λ	*maximal and minimal members of* $\mathcal{B}_\lambda$
$\Pi(\lambda)$	*the partition of* $X^\lambda - X_\lambda$ *induced by* $\mathcal{B}_\lambda$
Π_{pp}	*the partition of* S *obtained by taking the union of all the* $\Pi(\lambda)$
$(\Pi_{pp}, \geq_\pi)$	*partition partial order pair associated with* $(f(\cdot), g(\cdot))$
$\emptyset, E_1, \cdots, E_t$	*(usually) the principal sequence of* $(f(\cdot), g(\cdot))$
$\lambda_1, \cdots, \lambda_t$	*(usually) decreasing sequence of critical values*
$(\geq_R)$	*refined partial order associated with* $(f(\cdot), g(\cdot))$

Truncation and PLP

$\overline{f}(\Pi)$	$\sum_{N_i\in\Pi} f(N_i)$
$f_t(\cdot)$	$f_t(\emptyset)\equiv 0,$ $f_t(X)\equiv min_{\Pi\in\mathcal{P}_X}(\sum_{X_i\in\Pi} f(X_i))$
$\mathcal{L}_{\lambda_f}, f(\cdot)$ *on* 2^S	*collection of all partitions of* S *that minimize* $\overline{f-\lambda}(\cdot)$
$\mathcal{L}_\lambda$	$\mathcal{L}_{\lambda_f}$
Π^λ, Π_λ	*maximal and minimal member partitions in* $\mathcal{L}_\lambda$
$\lambda_1, \cdots, \lambda_t$	*(usually) decreasing sequence of critical PLP values of* $f(\cdot)$
$\Pi_{\lambda_1}, \Pi_{\lambda_2}, \cdots, \Pi_{\lambda_t}, \Pi^{\lambda_t}$	*principal sequence of partitions of* $f(\cdot)$
$\Pi'_{fus\cdot\Pi}, \Pi' \geq \Pi$	*partition of* Π *with* N_{fus} *as one of its blocks iff the members of* N_{fus} *are the set of blocks of* Π

	contained in a single block of Π'
$(\Pi_{fus})_{exp.\Pi}$	$(\Pi_{fus}$, *a partition of* $\Pi)$ *a partition with* N *as a block, iff* N *is the union of all blocks of* Π *which are members of a single block of* Π_{fus}

Contents

Chapter 1

Mathematical Preliminaries

1.1 Sets

A **set** (or **collection**) is specified by the **elements** (or **members**) that **belong** to it. If element x belongs to the set (does not belong to the set) X, we write $x \in X$ ($x \notin X$). Two sets are equal iff they have the same members. The set with no elements is called the empty set and is denoted by $\emptyset$. A set is **finite** if it has a finite number of elements. Otherwise it is **infinite**. A set is often specified by actually listing its members, e.g. $\{e_1, e_2, e_3\}$ is the set with members e_1, e_2, e_3. More usually it is specified by a property, e.g. the set of even numbers is specified as $\{x : x \text{ is an integer and } x \text{ is even }\}$ or as $\{x, x \text{ is an integer and } x \text{ is even }\}$. The symbols $\forall$ and $\exists$ are used to denote 'forall' and 'there exists'. Thus, '$\forall x$' or '$\forall\ x$' should be read as 'forall x' and '$\exists x$' should be read as 'there exists x'. A singleton set is one that has only one element. The singleton set with the element x as its only member, is denoted by $\{x\}$. In this book, very often, we abuse this notation and write x in place of $\{x\}$, if we feel that the context makes the intended object unambiguous.

We say that set X is **contained** in Y (**properly contained in** Y), if every element of X is also a member of Y (every element of X is a member of Y and $X \neq Y$) and denote it by $X \subseteq Y (X \subset Y)$.
The **union** of two sets X and Y denoted by $X \cup Y$, is the set whose members are either in X or in Y (or in both). The **intersection** of

X and Y denoted by $X \cap Y$, is the set whose members belong both to X and to Y. When X and Y do not have common elements, they are said to be **disjoint**. Union of disjoint sets X and Y is often denoted by $X \uplus Y$. Union of sets $X_1, \cdots, X_n$ is denoted by $\bigcup_{i=1}^n X_i$ or simply by $\bigcup X_i$. When the X_i are pairwise disjoint, their union is denoted by $\biguplus_{i=1}^n X_i$ or $\biguplus X_i$.
The **difference** of X relative to Y, denoted by $X - Y$, is the set of all elements in X but not in Y. Let $X \subseteq S$. Then the **complement** of X **relative to** S is the set $S - X$ and is denoted by $\bar{X}$ when the set S is clear from the context.

A **mapping** $f : X \to Y$, denoted by $f(\cdot)$, associates with each element $x \in X$, the element $f(x)$ in Y. The element $f(x)$ is called the image of x under $f(\cdot)$. We say $f(\cdot)$ maps X into Y. The sets X, Y are called, respectively, the **domain** and **codomain** of $f(\cdot)$. We denote by $f(Z), Z \subseteq X$, the subset of Y which has as members, the images of elements in Z. The set $f(X)$ is called the **range** of $f(\cdot)$. The **restriction** of $f(\cdot)$ to $Z \subseteq X$, denoted by $f/Z(\cdot)$ is the mapping from Z to Y defined by $f/Z(x) \equiv f(x), x \in Z$. A mapping that has distinct images for distinct elements in the domain is said to be **one to one** or **injective**. If the range of $f(\cdot)$ equals its codomain, we say that $f(\cdot)$ is **onto** or **surjective**. If the mapping is one to one onto we say it is **bijective**. Let $f : X \to Y, g : Y \to Z$. Then the **composition** of g and f is the map, denoted by $gf(\cdot)$ or $g \circ f(\cdot)$, defined by $gf(x) \equiv g(f(x)) \ \forall x \in X$. The **Cartesian product** $X \times Y$ of sets X, Y is the collection of all ordered pairs (x, y), where $x \in X$ and $y \in Y$. The **direct sum** $X \oplus Y$ denotes the union of disjoint sets X, Y. We use '**direct sum**' loosely to indicate that structures on two disjoint sets are 'put together'. We give some examples where we anticipate definitions which would be given later. The **direct sum** of vector spaces $\mathcal{V}_1, \mathcal{V}_2$ on disjoint sets S_1, S_2 is the vector space $\mathcal{V}_1 \oplus \mathcal{V}_2$ on $S_1 \oplus S_2$ whose typical vectors are obtained by taking a vector $\mathbf{x}_1 \equiv (a_1, \cdots, a_k)$ in $\mathcal{V}_1$ and a vector $\mathbf{x}_2 \equiv (b_1, \cdots, b_m)$ in $\mathcal{V}_2$ and putting them together as $\mathbf{x}_1 \oplus \mathbf{x}_2 \equiv (a_1, \cdots, a_k, b_1, \cdots, b_m)$. When we have two graphs $\mathcal{G}_1, \mathcal{G}_2$ on disjoint edge sets E_1, E_2, $\mathcal{G}_1 \oplus \mathcal{G}_2$ would have edge set $E_1 \oplus E_2$ and is obtained by 'putting together' $\mathcal{G}_1$ and $\mathcal{G}_2$. Usually the vertex sets would also be disjoint. However, where the context permits, we may relax the latter assumption and allow 'hinging' of vertices.

We speak of a **family** of subsets as distinct from a **collection** of subsets. The collection $\{\{e_1, e_2\}, \{e_1, e_2\}, \{e_1\}\}$ is identical to $\{\{e_1, e_2\}, \{e_1\}\}$. But often (e.g. in the definition of a hypergraph in Subsection 2.6.6) we have to use copies of the same subset many times and distinguish between copies. This we do by 'indexing' them. A family of subsets of S may be defined to be a mapping from an index set I to the collection of all subsets of S. For the purpose of this book, the index set I can be taken to be $\{1, \cdots, n\}$. So the family $(\{e_1, e_2\}, \{e_1, e_2\}, \{e_1\})$ can be thought of as the mapping $\phi(\cdot)$ with

$$\begin{aligned} \phi(1) &\equiv \{e_1, e_2\} \\ \phi(2) &\equiv \{e_1, e_2\} \\ \phi(3) &\equiv \{e_1\}. \end{aligned}$$

(Note that a family is denoted using ordinary brackets while a set is denoted using curly brackets).

1.2 Vectors and Matrices

In this section we define vectors, matrices and related notions. Most present day books on linear algebra treat vectors as primitive elements in a vector space and leave them undefined. We adopt a more old fashioned approach which is convenient for the applications we have in mind. The reader who wants a more leisurely treatment of the topics in this section is referred to [Hoffman+Kunze72].

Let S be a finite set $\{e_1, e_2, \ldots, e_n\}$ and let $\mathcal{F}$ be a field. We will confine ourselves to the field $\Re$ of real numbers, the field $\mathcal{C}$ of complex numbers and the $GF2$ field on elements $0, 1$ $(0+0 = 0, 0+1 = 1, 1+0 = 1, 1 + 1 = 0, 1.1 = 1, 1.0 = 0, 0.1 = 0, 0.0 = 0)$. For a general definition of a field see for instance [Jacobson74]. By a vector on S over $\mathcal{F}$ we mean a mapping $\mathbf{f}$ of S into $\mathcal{F}$.The field $\mathcal{F}$ is called the **scalar field** and its elements are called **scalars**. The **support** of $\mathbf{f}$ is the subset of S over which it takes nonzero values. The **sum** of two vectors $\mathbf{f}, \mathbf{g}$ on S over $\mathcal{F}$ is defined by $(\mathbf{f} + \mathbf{g})(e_i) \equiv \mathbf{f}(e_i) + \mathbf{g}(e_i)\ \forall e_i \in S$. (For convenience the **sum** of two vectors $\mathbf{f}$ on S,$\mathbf{g}$ on T over $\mathcal{F}$ is defined by $(\mathbf{f}+\mathbf{g})(e_i) \equiv \mathbf{f}(e_i)+\mathbf{g}(e_i)\ \forall e_i \in S\cap T$, as agreeing with $\mathbf{f}$ on $S-T$, and as agreeing with $\mathbf{g}$ on $T - S$). The **scalar product** of $\mathbf{f}$ by a scalar λ

is a vector $\lambda\mathbf{f}$ defined by $(\lambda\mathbf{f})(e_i) \equiv \lambda(\mathbf{f}(e_i))\ \forall e_i \in S$. A collection $\mathcal{V}$ of vectors on S over $\mathcal{F}$ is a vector space iff it is closed under addition and scalar product. We henceforth omit mention of underlying set and field unless required.

A set of vectors $\{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_n\}$ is **linearly dependent** iff there exist scalars $\lambda_1, \ldots, \lambda_n$ not all zero such that $\lambda_1\mathbf{f}_1 + \ldots + \lambda_n\mathbf{f}_n = \mathbf{0}$. (Here the $\mathbf{0}$ vector is one which takes value 0 on all elements of S). Vector $\mathbf{f}_n$ is a linear combination of $\mathbf{f}_1, \ldots, \mathbf{f}_{n-1}$ iff $\mathbf{f}_n = \lambda_1\mathbf{f}_1 + \ldots + \lambda_{n-1}\mathbf{f}_{n-1}$ for some $\lambda_1, \ldots, \lambda_{n-1}$.

The set of all vectors linearly dependent on a collection $\mathcal{C}$ of vectors can be shown to form a vector space which is said to be **generated** by or **spanned** by $\mathcal{C}$. Clearly if $\mathcal{V}$ is a vector space and $\mathcal{C} \subseteq \mathcal{V}$, the subset of vectors generated by $\mathcal{C}$ is contained in $\mathcal{V}$. A maximal linearly independent set of vectors of $\mathcal{V}$ is called a **basis** of $\mathcal{V}$.
In general maximal and minimal members of a collection of sets may not be largest and smallest in terms of size.
Example: Consider the collection of sets $\{\{1,2,3\},\{4\},\{5,6\},\{1,2,3,5,6\}\}$. The minimal members of this collection are $\{1,2,3\},\{4\},\{5,6\}$, i.e., these do not contain proper subsets which are members of this collection. The maximal members of this collection are $\{4\},\{1,2,3,5,6\}$, i.e., these are not proper subsets of other sets which are members of this collection.

The following theorem is therefore remarkable.

Theorem 1.2.1 *All bases of a vector space on a finite set have the same cardinality.*

The number of elements in a basis of $\mathcal{V}$ is called the **dimension** of $\mathcal{V}$, denoted by $dim(\mathcal{V})$, or the **rank** of $\mathcal{V}$, denoted by $r(\mathcal{V})$. Using Theorem 1.2.1 one can show that the size of a maximal independent subset contained in a given set $\mathcal{C}$ of vectors is unique. This number is called the **rank** of $\mathcal{C}$. Equivalently, the rank of $\mathcal{C}$ is the dimension of the vector space spanned by $\mathcal{C}$. If $\mathcal{V}_1, \mathcal{V}_2$ are vector spaces and $\mathcal{V}_1 \subseteq \mathcal{V}_2$, we say $\mathcal{V}_1$ is a **subspace** of $\mathcal{V}_2$.

A mapping $\mathbf{A} : \{1,2,\ldots,m\} \times \{1,2,\ldots,n\} \longrightarrow \mathcal{F}$ is called a $m \times n$ matrix. It may be thought of as an $m \times n$ array with entries from $\mathcal{F}$. We denote $\mathbf{A}(i,j)$ often by the lower case a_{ij} with i as the row index and j as the column index. We speak of the array $(a_{i1}, \ldots, a_{in})$ as the

ith row of $\mathbf{A}$ and of the array $(a_{1j}, \ldots, a_{nj})$ as the **jth column** of $\mathbf{A}$. Thus we may think of $\mathbf{A}$ as made up of m row vectors or of n column vectors. Linear dependence, independence and linear combination for row and column vectors are defined the same way as for vectors. We say two matrices are **row equivalent** if the rows of each can be obtained by linearly combining the rows of the other. **Column equivalence** is defined similarly. The vector space spanned by the rows (columns) of $\mathbf{A}$ is called its **row space (column space)** and denoted by $\mathcal{R}(\mathbf{A})(\mathcal{C}(\mathbf{A}))$. The dimension of $\mathcal{R}(\mathbf{A})(\mathcal{C}(\mathbf{A}))$ is called the **row rank (column rank)** of $\mathbf{A}$.

If $\mathbf{A}$ is an $m \times n$ matrix then the **transpose** of $\mathbf{A}$ denoted by $\mathbf{A}^T$ is an $n \times m$ matrix defined by $\mathbf{A}^T(i,j) \equiv \mathbf{A}(j,i)$. Clearly the i^{th} row of $\mathbf{A}$ becomes the i^{th} column of $\mathbf{A}^T$ and vice versa. If $\mathbf{B}$ is also an $m \times n$ matrix the **sum** $\mathbf{A}+\mathbf{B}$ is an $m \times n$ matrix defined by $(\mathbf{A}+\mathbf{B})(i,j) \equiv \mathbf{A}(i,j)+\mathbf{B}(i,j)$. If $\mathbf{D}$ is an $n \times p$ matrix, the product $\mathbf{AD}$ is an $m \times p$ matrix defined by $\mathbf{AD}(i,j) \equiv \sum_{k=1}^{n} a_{ik}d_{kj}$. Clearly if $\mathbf{AD}$ is defined it does not follow that $\mathbf{DA}$ is defined. Even when it is defined, in general $\mathbf{AD} \neq \mathbf{DA}$. The most basic property of this notion of product is that it is **associative** i.e. $\mathbf{A}(\mathbf{DF}) = (\mathbf{AD})\mathbf{F}$.

Matrix operations are often specified by **partitioning**. Here we write a matrix in terms of **submatrices** (i.e., matrices obtained by deleting some rows and columns of the original matrix). A matrix may be partitioned along rows:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} \\ \ldots \\ \mathbf{A}_{21} \end{bmatrix}$$

or along columns:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} | \mathbf{A}_{12} \end{bmatrix}$$

or both:

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_{11} & \ldots & \mathbf{A}_{1k} \\ \vdots & \vdots & \vdots \\ \mathbf{A}_{p1} & \ldots & \mathbf{A}_{pk} \end{bmatrix}.$$

When two partitioned matrices are multiplied we assume that the partitioning is **compatible**, i.e., for each triple (i,j,k) the number of columns of $\mathbf{A}_{ik}$ equals the number of rows of $\mathbf{B}_{kj}$. Clearly this is achieved if the original matrices $\mathbf{A}, \mathbf{B}$ are compatible for product and each block of the column partition of $\mathbf{A}$ has the same size as the corresponding row partition of $\mathbf{B}$. The following **partitioning rules** can then be verified.

i. $\begin{bmatrix} \mathbf{A}_{11} \\ \dots \\ \mathbf{A}_{21} \end{bmatrix} \mathbf{C} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{C} \\ \dots \\ \mathbf{A}_{21}\mathbf{C} \end{bmatrix}$

ii. $\mathbf{C} \begin{bmatrix} \mathbf{A}_{11} | \mathbf{A}_{12} \end{bmatrix} = \begin{bmatrix} \mathbf{C}\mathbf{A}_{11} | \mathbf{C}\mathbf{A}_{12} \end{bmatrix}$

iii. $\begin{bmatrix} \mathbf{A}_{11} | \mathbf{A}_{12} \end{bmatrix} \begin{bmatrix} \mathbf{C}_{11} \\ \dots \\ \mathbf{C}_{12} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{11}\mathbf{C}_{11} + \mathbf{A}_{12}\mathbf{C}_{12} \end{bmatrix}.$

In general if $\mathbf{A}$ is partitioned into submatrices $\mathbf{A}_{ik}, \mathbf{B}$ into submatrices $\mathbf{B}_{kj}$ then the product $\mathbf{C} = \mathbf{AB}$ would be naturally partitioned into $\mathbf{C}_{ij} \equiv \sum_k \mathbf{A}_{ik}\mathbf{B}_{kj}$.

Matrices arise most naturally in linear equations such as $\mathbf{Ax} = \mathbf{b}$, where $\mathbf{A}$ and $\mathbf{b}$ are known and $\mathbf{x}$ is an unknown vector. When $\mathbf{b} = \mathbf{0}$ it is easily verified that the set of all solutions of $\mathbf{Ax} = \mathbf{b}$, i.e.,of $\mathbf{Ax} = \mathbf{0}$, forms a vector space. This space will be called the **solution space** of $\mathbf{Ax} = \mathbf{0}$, or the **null space** of $\mathbf{A}$. The **nullity** of $\mathbf{A}$ is the dimension of the null space of $\mathbf{A}$. We have the following theorem.

Theorem 1.2.2 *If two matrices are row equivalent then their null spaces are identical.*

Corollary 1.2.1 *If* $\mathbf{A}, \mathbf{B}$ *are row equivalent matrices then a set of columns of* $\mathbf{A}$ *is independent iff the corresponding set of columns of* $\mathbf{B}$ *is independent.*

The following are **elementary row operations** that can be performed on the rows of a matrix:

i. interchanging rows,

ii. adding a multiple of one row to another,

iii. multiplying a row by a nonzero number.

Each of these operations corresponds to premultiplication by a matrix. Such matrices are called **elementary matrices**. It can be seen that these are the matrices we obtain by performing the corresponding elementary row operations on the unit matrix of the same number of rows as the given matrix. We can define elementary column operations similarly. These would correspond to post multiplication by elementary column matrices.

A matrix is said to be in **Row Reduced Echelon** form (RRE) iff it satisfies the following:
Let r be the largest row index for which $a_{ij} \neq 0$ for some j. Then the columns of the $r \times r$ unit matrix (the matrix with $1s$ along the diagonal and zero elsewhere) $\mathbf{e_1}, \ldots, \mathbf{e_r}$ appear as columns, say $\mathbf{C}_{i_1}, \ldots, \mathbf{C}_{i_r}$ of $\mathbf{A}$ with $i_1 < \ldots < i_r$. Further if $p < i_k$ then $a_{kp} = 0$. We have the following theorem.

Theorem 1.2.3 *Every matrix can be reduced to a matrix in the RRE form by a sequence of elementary row transformations and is therefore row equivalent to such a matrix.*

It is easily verified that for an RRE matrix row rank equals column rank. Hence using Theorem 1.2.3 and Corollary 1.2.1 we have

Theorem 1.2.4 *For any matrix, row rank equals column rank.*

The **rank** of a matrix $\mathbf{A}$, denoted by $r(\mathbf{A})$, is its row rank (= column rank).
Let the elements of S be ordered as $(e_1, \ldots, e_n)$. Then for any vector $\mathbf{f}$ on S we define $\mathbf{R}_f$, the **representative vector** of $\mathbf{f}$, as the one rowed matrix $(\mathbf{f}(e_1), \ldots, \mathbf{f}(e_n))$. We will not usually distinguish between a vector and its representative vector. When the rows of a matrix $\mathbf{R}$ are representative vectors of some basis of a vector space $\mathcal{V}$ we say that $\mathbf{R}$ is a **representative matrix** of $\mathcal{V}$. When $\mathbf{R}, \mathbf{R}_1$ both represent $\mathcal{V}$ they can be obtained from each other by row operations. Hence by Corollary 1.2.1 their column independence structure is identical. An $r \times n$ representative matrix $\mathbf{R}$, $r \leq n$, is a **standard representative matrix** iff $\mathbf{R}$ has an $r \times r$ submatrix which can be obtained by permutations of the columns of the $r \times r$ unit matrix. For convenience we will write a standard representative matrix in the form $[\mathbf{I}|\mathbf{R}_{12}]$ or $[\mathbf{R}_{11}|\mathbf{I}]$. (Here $\mathbf{I}$ denotes the unit matrix of appropriate size).

The **dot product** of two vectors $\mathbf{f}, \mathbf{g}$ on S denoted by $< \mathbf{f}, \mathbf{g} >$ over

$\mathcal{F}$ is defined by $< \mathbf{f}, \mathbf{g} > \equiv \sum_{e \in S} \mathbf{f}(e).\mathbf{g}(e)$. We say $\mathbf{f}, \mathbf{g}$ are **orthogonal** if their dot product is zero. If $\mathcal{C}$ is a collection of vectors on S then $\mathcal{C}^{\perp} \equiv$ set of all vectors orthogonal to every vector in $\mathcal{C}$. It can be verified that $\mathcal{C}^{\perp}$ is a vector space. Let $\mathcal{V}$ be a vector space on S with basis $\mathcal{B}$. Since vectors orthogonal to each vector in $\mathcal{B}$ are also orthogonal to linear combinations of these vectors we have $\mathcal{B}^{\perp} = \mathcal{V}^{\perp}$. If $\mathbf{R}$ is a representative matrix of $\mathcal{V}$, it is clear that $\mathcal{V}^{\perp}$ is its null space. Equivalently $\mathcal{V}^{\perp}$ is the solution space of $\mathbf{Rx} = \mathbf{0}$. If $\mathbf{R}$ is a standard representative matrix with $\mathbf{R} = [\mathbf{I}_{r \times r} | \mathbf{R}_{12}]$, then the solution space of $\mathbf{Rx} = \mathbf{0}$ can be shown to be the vector space generated by the columns of $\begin{bmatrix} -\mathbf{R}_{12} \\ \cdots \\ \mathbf{I}_{n-r \times n-r} \end{bmatrix}$, where $n = |S|$.(Here $\mathbf{I}_{k \times k}$ denotes the unit matrix with k rows). Equivalently $\mathcal{V}^{\perp}$ has the representative matrix $[-\mathbf{R}_{12}^T | \mathbf{I}_{n-r \times n-r}]$. The representative matrix of $(\mathcal{V}^{\perp})^{\perp}$ will then be $\mathbf{R}$. We then have the following

Theorem 1.2.5 *i. if* $[\mathbf{I}_{r \times r} | \mathbf{R}_{12}]$ *is a representative matrix of vector space* $\mathcal{V}$ *on* S *then* $[-\mathbf{R}_{12}^T | \mathbf{I}_{n-r \times n-r}]$ *is a representative matrix of* $\mathcal{V}^{\perp}$.

ii. $r(\mathcal{V}^{\perp}) = |S| - r(\mathcal{V})$

iii. $(\mathcal{V}^{\perp})^{\perp} = \mathcal{V}$. *Hence two matrices are row equivalent iff their null spaces are identical.*

Consider the collection of all $n \times n$ matrices over $\mathcal{F}$. We say that $\mathbf{I}$ is an identity for this collection iff for every $n \times n$ matrix $\mathbf{B}$ we have $\mathbf{IB} = \mathbf{BI} = \mathbf{B}$. If $\mathbf{I}_1, \mathbf{I}_2$ are identity matrices we must have $\mathbf{I}_1 = \mathbf{I}_2 = \mathbf{I}$. The unit matrix (with $1s$ along the diagonal and $0s$ elsewhere) is clearly an identity matrix. It is therefore the only identity matrix. Two $n \times n$ matrices $\mathbf{A}, \mathbf{B}$ are **inverses** of each other iff $\mathbf{AB} = \mathbf{BA} = \mathbf{I}$. We say $\mathbf{A}, \mathbf{B}$ are **invertible** or **nonsingular**. If $\mathbf{A}$ has inverses $\mathbf{B}, \mathbf{C}$ we must have $\mathbf{C} = \mathbf{C}(\mathbf{AB}) = (\mathbf{CA})\mathbf{B} = \mathbf{IB} = \mathbf{B}$. Thus the inverse of a matrix $\mathbf{A}$, if it exists, is unique and is denoted by $\mathbf{A}^{-1}$. We then have the following

Theorem 1.2.6 *i.* $(\mathbf{A}^T)^{-1} = (\mathbf{A}^{-1})^T$

ii. If $\mathbf{A}, \mathbf{D}$ *are* $n \times n$ *invertible matrices, then* $(\mathbf{AD})^{-1} = (\mathbf{D}^{-1}\mathbf{A}^{-1})$.

With a square matrix we associate an important number called its determinant. Its definition requires some preparation.

A bijection of a finite set to itself is also called a **permutation**. A permutation that interchanges two elements (i.e. maps each of them to the other) but leaves all others unchanged is a **transposition**. Every permutation can be obtained by repeated application of transpositions. We then have the following

Theorem 1.2.7 *If a permutation σ can be obtained by composition of an even number of transpositions then every decomposition of σ into transpositions will contain an even number of them.*

By Theorem 1.2.7 we can define a permutation to be **even (odd)** iff it can be decomposed into an even (odd) number of transpositions. The **sign** of a permutation σ denoted by $sgn(\sigma)$ is $+1$ if σ is even and -1 if σ is odd. It is easily seen, since the identity ($= \sigma\sigma^{-1}$) permutation is even, that $sgn(\sigma) = sgn(\sigma^{-1})$. The **determinant** of an $n \times n$ matrix is defined by

$$det(\mathbf{A}) \equiv \sum_{\sigma} sgn(\sigma) a_{1\sigma(1)} \ldots a_{n\sigma(n)},$$

where the summation is taken over all possible permutations of $\{1, 2, \ldots, n\}$. It is easily seen that determinant of the unit matrix is $+1$. We collect some of the important properties of the determinant in the following

Theorem 1.2.8 *i.* $det(\mathbf{A}) = det(\mathbf{A}^T)$

ii. *Let*

$$\mathbf{A} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{A}_2 \end{bmatrix}, \mathbf{A}' = \begin{bmatrix} \mathbf{a}'_1 \\ \mathbf{A}_2 \end{bmatrix}, \mathbf{A}" = \begin{bmatrix} \mathbf{a}_1 + \mathbf{a}'_1 \\ \mathbf{A}_2 \end{bmatrix}.$$

Then $det(\mathbf{A}") = det(\mathbf{A}) + det(\mathbf{A}')$.

iii. *If* $\mathbf{A}$ *has two identical rows, or has two identical columns then* $det(\mathbf{A}) = 0$.

iv. *If* $\mathbf{E}$ *is an elementary matrix* $det(\mathbf{EA}) = det(\mathbf{E})det(\mathbf{A})$. *Since every invertible matrix can be factored into elementary matrices, it follows that* $det(\mathbf{AB}) = det(\mathbf{A})det(\mathbf{B})$, *for every pair of* $n \times n$ *matrices* $\mathbf{A}, \mathbf{B}$.

v. $det(\mathbf{A}) \neq 0$ *iff* $\mathbf{A}$ *is invertible.*

Problem 1.1 Size of a basis: *Prove*

i. *Theorem 1.2.1*

ii. *If $\mathcal{V}_1$ is a subspace of vector space $\mathcal{V}_2$, $\dim \mathcal{V}_1 \leq \dim \mathcal{V}_2$.*

iii. *If $\mathcal{V}_1 \subseteq \mathcal{V}_2$ and $\dim \mathcal{V}_1 = \dim \mathcal{V}_2$ then $\mathcal{V}_1 = \mathcal{V}_2$.*

iv. *an $m \times n$ matrix with $m > n$ cannot have linearly independent rows.*

v. *any vector in a vector space $\mathcal{V}$ can be written uniquely as a linear combination of the vectors in a basis of $\mathcal{V}$.*

Problem 1.2 Ways of interpreting the matrix product: *Define product of matrices in the usual way i.e. $\mathbf{C} = \mathbf{AB}$ is equivalent to $\mathbf{C}_{ij} = \sum_k a_{ik} b_{kj}$. Now show that it can be thought of as follows*

i. *Columns of $\mathbf{C}$ are linear combinations of columns of $\mathbf{A}$ using entries of columns of $\mathbf{B}$ as coefficients.*

ii. *rows of $\mathbf{C}$ are linear combinations of rows of $\mathbf{B}$ using entries of rows of $\mathbf{A}$ as coefficients.*

Problem 1.3 Properties of matrix product: *Prove, when* **A, B, C** *are matrices and the products are defined*

i. $(\mathbf{AB})\mathbf{C} = \mathbf{A}(\mathbf{BC})$

ii. $(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$

Problem 1.4 Partitioning rules: *Prove*

i. *the partitioning rules.*

ii.

$$\begin{bmatrix} \mathbf{A}_{11} \cdots & \mathbf{A}_{1k} \\ \vdots & \vdots \\ \mathbf{A}_{r1} \cdots & \mathbf{A}_{rk} \end{bmatrix}^T = \begin{bmatrix} \mathbf{A}_{11}^T \cdots & \mathbf{A}_{r1}^T \\ \vdots & \vdots \\ \mathbf{A}_{1k}^T \cdots & \mathbf{A}_{rk}^T \end{bmatrix}$$

Problem 1.5 Solution space and column dependence structure: *Prove theorem 1.2.2 and Corollary 1.2.1.*

Problem 1.6 Algorithm for computing RRE: *Give an algorithm for converting any rectangular matrix into the RRE form. Give an upper bound for the number of arithmetical steps in your algorithm.*

Problem 1.7 Uniqueness of the RRE matrix: *Show that no RRE matrix is row equivalent to a distinct RRE matrix. Hence prove that every matrix is row equivalent to a unique matrix in the RRE form.*

Problem 1.8 RRE of special matrices:

i. *If* $\mathbf{A}$ *is a matrix with linearly independent columns what is its RRE form? If in addition* $\mathbf{A}$ *is square what is its RRE form?*

ii. *If* $\mathbf{A}, \mathbf{B}$ *are square such that* $\mathbf{AB} = \mathbf{I}$ *show that* $\mathbf{BA} = \mathbf{I}$.

iii. *Prove Theorem 1.2.6*

Problem 1.9 Existence and nature of solution for linear equations: *Consider the equation* $\mathbf{Ax} = \mathbf{b}$.

i. *Show that it has a solution*

(a) *iff* $r(\mathbf{A}) = r(\mathbf{A}|\mathbf{b})$.

(b) *iff whenever* $\lambda^T\mathbf{A} = \mathbf{0}, \lambda^T\mathbf{b}$ *is also zero.*

ii. *Show that a vector is a solution of the above equation iff it can be written in the form* $\mathbf{x}_o + \mathbf{x}_p$ *where* $\mathbf{x}_p$ *is a particular solution of the equation while* $\mathbf{x}_o$ *is a vector in the null space of* $\mathbf{A}$ *(i.e. a solution to the linear equation with* $\mathbf{b}$ *set equal to zero).*

iii. **Motivation for the matrix product:** *Why is the matrix product defined as in Problem 1.2? (In the above equation suppose we make the substitution* $\mathbf{x} = \mathbf{By}$. *What would the linear equation in terms of* $\mathbf{y}$ *be?)*

iv. **Linear dependence and logical consequence:** *The above equation may be regarded as a set of linear equations (one for each row of* $\mathbf{A}$*) each of which in turn could be thought of as a statement. Show that a linear equation is a logical consequence of others iff it is* **linearly** *dependent on the others.*

Problem 1.10 Positive definite matrices:

i. *Construct an example where* $\mathbf{A}$, $\mathbf{B}$ *are invertible but their sum is not.*

ii. *A matrix* $\mathbf{K}$ *is* **positive semidefinite (positive definite)** *iff* $\mathbf{x}^T\mathbf{K}\mathbf{x} \geq 0 \quad \forall \mathbf{x} \neq 0$ *(*$\mathbf{x}^T\mathbf{K}\mathbf{x} > 0 \quad \forall \mathbf{x} \neq 0$*). Show that*

   (a) *a matrix is invertible if it is positive definite;*

   (b) *sum of two positive semidefinite matrices (positive definite matrices) is positive semidefinite (positive definite);*

   (c) *if* $\mathbf{K}$ *is a positive definite matrix,then* $\mathbf{AKA}^T$ *is positive semidefinite and if, further, rows of* $\mathbf{A}$ *are linearly independent, then* $\mathbf{AKA}^T$ *is positive definite;*

   (d) *inverse of a symmetric positive definite matrix is also symmetric positive definite.*

Problem 1.11 Projection of a vector on a vector space: *Let* $\mathbf{x}$ *be a vector on* S *and let* $\mathcal{V}$ *be a vector space on* S*. Show that* $\mathbf{x}$ *can be uniquely decomposed as* $\mathbf{x} = \mathbf{x}_1 + \mathbf{x}_2$*, where* $\mathbf{x}_1 \in \mathcal{V}$ *and* $\mathbf{x}_2 \in \mathcal{V}^\perp$*.The vector* $\mathbf{x}_1$ *is called the* **projection** *of* $\mathbf{x}$ *on* $\mathcal{V}$ **along** $\mathcal{V}^\perp$*.*

Problem 1.12 Parity of a Permutation: *Show that if a permutation can be obtained by composing an odd number of transpositions it cannot also be obtained by composing an even number of transpositions.*

Problem 1.13 Graph of a permutation: *Define the graph* $\mathcal{G}_\sigma$ *of a permutation* σ *on* $\{1, 2, \cdots n\}$ *as follows:* $V(\mathcal{G}_\sigma) \equiv \{1, 2, \cdots, n\}$*; draw an edge with an arrow from* i *to* j *iff* $\sigma(i) = j$*.*

i. *Show that every vertex in this graph has precisely one arrow coming in and one going out. Hence, conclude that each connected component is a directed circuit.*

ii. *Show that if* $\mathcal{G}_\sigma$ *has an odd (even) number of even length circuits then* σ *is odd (even).*

Problem 1.14 Properties of the determinant: *Prove Theorem 1.2.8.*

Problem 1.15 Equivalence of definitions of a determinant: *Show that the usual definition of a determinant by expanding along a row or column is equivalent to the definition using permutations.*

Problem 1.16 Laplace expansion of the determinant: *Let* **A** *be an* $n \times n$ *matrix. Show that*

$$det(\mathbf{A}) = \sum sgn(\sigma)\ det\left(\mathbf{A}\begin{pmatrix} r_1, & \cdots & , r_k \\ i_1, & \cdots & , i_k \end{pmatrix}\right) det\left(\mathbf{A}\begin{pmatrix} r_{k+1}, & \cdots & , r_m \\ i_{k+1}, & \cdots & , i_m \end{pmatrix}\right),$$

($\mathbf{A}\begin{pmatrix} d_1, & \cdots & , d_p \\ i_1, & \cdots & , i_p \end{pmatrix}$ *is the* $p \times p$ *matrix whose* (s,t) *entry is the* (d_s, i_t) *entry of* **A***), where the summation is over all subsets* $\{r_1, \cdots, r_k\}$ *of* $\{1, \cdots, n\}$
and $\sigma \equiv \begin{pmatrix} r_1, \cdots, r_k, r_{k+1} \cdots r_n \\ i_1, \cdots, i_k, i_{k+1} \cdots i_n \end{pmatrix}$ *i.e.,* $\sigma(r_j) \equiv i_j, j = 1, \cdots, n.$

Problem 1.17 Binet Cauchy Theorem: *Let* **A** *be an* $m \times n$ *and* **B** *an* $n \times m$ *matrix with* $m \leq n$. *If an* $m \times m$ *submatrix of* **A** *is composed of columns* $i_1, \cdots, i_m$, *the* **corresponding** $m \times m$ *submatrix of* **B** *is the one with rows* $i_1, \cdots, i_m$. *Prove the Binet Cauchy Theorem:* $det(\mathbf{AB}) =$ $\sum$ *product of determinants of corresponding* $m \times m$ *submatrices of* **A** *and* **B**.

1.3 Linear Inequality Systems

1.3.1 The Kuhn-Fourier Theorem

In this section we summarize basic results on inequality systems which we need later on in the book. Proofs are mostly omitted. They may be found in standard references such as [Stoer+Witzgall70] and [Schrijver86]. This section follows the former reference.

A **linear inequality system** is a set of constraints of the following kind on the vector $\mathbf{x} \in \Re^n$.

$$\left.\begin{array}{rcl} \mathbf{Ax} & = & \mathbf{a}_o \\ \mathbf{Bx} & > & \mathbf{b}_o \\ \mathbf{Cx} & \geq & \mathbf{c}_o \end{array}\right\} \qquad (I)$$

Here, $\mathbf{A}, \mathbf{B}, \mathbf{C}$ are matrices, $\mathbf{a}_o, \mathbf{b}_o, \mathbf{c}_o$ are column vectors with appropriate number of rows. We say $\mathbf{x}_1 > \mathbf{x}_2 (\mathbf{x}_1 \geq \mathbf{x}_2)$ iff each component of $\mathbf{x}_1$ is greater than (greater than or equal to) the corresponding component of $\mathbf{x}_2$.

A **solution** of an inequality system is a vector which satisfies all the inequality constraints of the system. A constraint which is satisfied by every solution of an inequality system is said to be a **consequence** of the system. In particular, we are concerned with constraints of the kind $\mathbf{d}^T\mathbf{x} = \mathbf{d}_o$ or $> \mathbf{d}_o$ or $\geq \mathbf{d}_o$. A **legal linear combination** of the system (I) is obtained by linearly combining the equations and inequalities with real coefficients - α_i for the linear equations, and non-negative real coefficients β_j, γ_k for the '$>$' linear inequalities and '$\geq$' linear inequalities respectively. The resulting constraint would be a linear equation iff β_j, γ_k are all zero. It would be a '$>$' inequality iff at least one of the β_j's is nonzero. It would be a '$\geq$' inequality iff all of β_j are zero but at least one of the γ_k is nonzero. A legal linear combination is thus a consequence of the system. A legal linear combination, with at least one of the $\alpha_i, \beta_j, \gamma_k$ nonzero, that results in the LHS becoming zero is called a **legal linear dependence** of the system. Another important way of deriving consequence relations is by **weakening**. This means to weaken '$=$' to '$\geq$' and '$>$' to '$\geq$' and also in the case of '$>$' and '$\geq$' to lower the right side value.

Example 1.3.1 *Consider the system of linear inequalities:*

$$\begin{aligned} x_1 + 2x_2 &= 3 \\ 2x_1 + x_2 &= 4 \\ x_1 + x_2 &> 1 \\ 2x_1 + 3x_2 &> 2 \\ x_1 + 5x_2 &\geq 2 \\ -x_1 - 2x_2 &\geq 4. \end{aligned}$$

The legal linear combination corresponding to $\alpha_1 = 1, \alpha_2 = 1, \beta_1 = 0, \beta_2 = 0, \gamma_1 = 0, \gamma_2 = 0$ is
$3x_1 + 3x_2 = 7$;
that corresponding to $\alpha_1 = 1, \alpha_2 = 0, \beta_1 = 1, \beta_2 = 0, \gamma_1 = 1, \gamma_2 = 0$ is
$3x_1 + 8x_2 > 6$;
that corresponding to $\alpha_1 = 1, \alpha_2 = 0, \beta_1 = 0, \beta_2 = 0, \gamma_1 = 1, \gamma_2 = 0$ is
$2x_1 + 7x_2 \geq 5$.
The legal linear combination corresponding to $\alpha_1 = 1, \alpha_2 = 0, \beta_1 = 0, \beta_2 = 0, \gamma_1 = 0, \gamma_2 = 1$ is the zero relation
$0x_1 + 0x_2 \geq 7$.

Thus in this case, the system has a **legal linear dependence** that is a **contradiction**.

We can now state the fundamental theorem of Kuhn and Fourier [Fourier1826], [Kuhn56].

Theorem 1.3.1 (Kuhn-Fourier Theorem) *A linear inequality system has a solution iff no legal linear dependence is a contradiction.*

Sketch of the Proof of Theorem 1.3.1: First reduce the linear equations to the RRE form. If a row arises with zero coefficients but with nonzero right side at this stage, we have a legal linear dependence that is a contradiction. Otherwise express some of the variables in terms of the others. This substitution is now carried out also in the inequalities. So henceforth, without loss of generality, we may assume that we have only inequalities. If we prove the theorem for such a reduced system, it can be extended to one which has equalities also.

Suppose each variable has either zero coefficient or the same sign in all the inequalities of the system and further, if there are inequalities with zero coefficients they are not contradictory.

In this case it is easy to see that the system has a solution whether the coefficients of a particular variable are all zero or otherwise. If all the coefficients are zero we are done - the theorem is clearly true. If not, it is not possible to get a legal linear dependence without using zero coefficients. So the theorem is again true in this case.

We now present an elimination procedure which terminates at the above mentioned situation.
Let the inequalities be numbered $(1), \cdots, (r), (r+1), \cdots, (k)$. Let x_n be present with coefficient $+1$ in the inequalities $(1), \cdots, (r)$ and with coefficient -1 in the inequalities $(r+1), \cdots, (k)$. We create $r(k-r)$ inequalities without the variable x_n by adding each of the first r inequalities to each of the last $(k-r)$ inequalities. Note that if both the inequalities are of the $(\geq)$ kind, the addition would result in another of the $(\geq)$ kind and if one of them is of the $(>)$ kind, the addition would result in another of the $(>)$ kind.

If the original system has a solution, it is clear that the reduced system also has one. On the other hand, if the reduced system has a solution $(x'_1, \cdots, x'_{n-1})$ it is possible to find a value x'_n of x_n such that $(x'_1, \cdots, x'_{n-1}, x'_n)$ is a solution of the original system. We indicate how,

below.
Let the inequalities added be

$$a_{i1}x_1 + \cdots + x_n \geq b_i$$

$$a_{j1}x_1 + \cdots - x_n > b_j$$

(The cases where both are ($\geq$), both are ($>$) or first inequality ($>$) and second ($\geq$) are similar.) The pair of inequalities can be written equivalently as

$$a_{j1}x_1 + \cdots + a_{j(n-1)}x_{n-1} - b_j > x_n \geq b_i - a_{i1}x_1 - \cdots - a_{i(n-1)}x_{n-1} \qquad (*)$$

The extreme left of the above inequality ($*$) is always derived from the inequalities ($r+1$) to (k) while the extreme right is always derived from the (1) to (r) inequalities. When $x'_1, \cdots, x'_{n-1}$ is substituted in the above inequality, it would be satisfied for every pair of inequalities, from ($j+1$) to (k) on the extreme left and (1) to (j) on the extreme right. After substitution, let the least of the extreme left term be reached for inequality (p) and let the highest of the extreme right term be reached for inequality (q). Since

$$a_{p1}x'_1 + \cdots + a_{p(n-1)}x'_{n-1} - b_p > b_q - a_{q1}x'_1 - \cdots - a_{q(n-1)}x'_{n-1}$$

(this inequality results when (p) and (q) are added), we can find a value x'_n of x_n which lies between left and right sides of the above inequality. Clearly $(x'_1, \cdots, x'_n)$ is a solution of the original system.

If this procedure were repeated, we would reach a system where there are inequalities with all the coefficients of zero value and where the signs of the coefficients of a variable are all the same in all the inequalities. If some of the inequalities which have all zero coefficients are contradictory there is no solution possible and the theorem is true. If none of such inequalities are contradictory the solution always exists as mentioned before and there can be no legal linear combination that is contradictory. Thus once again the theorem is true.

□

As an immediate consequence we can prove the celebrated 'Farkas Lemma'.

Theorem 1.3.2 (Farkas Lemma) *The homogeneous system*

$$\mathbf{A}\ \mathbf{x} \leq 0$$

has the consequence

$$\mathbf{d}^T\mathbf{x} \leq 0$$

iff the row vector $\mathbf{d}^T$ *is a nonnegative linear combination of the rows of* $\mathbf{A}$.

Proof : By Kuhn-Fourier Theorem (Theorem 1.3.1), the system

$$\mathbf{A}^T\mathbf{y} = \mathbf{d}$$

$$\mathbf{y} \geq \mathbf{0}$$

has a solution iff

‘$\mathbf{x}^T\mathbf{A}^T + \beta^T\mathbf{I} = \mathbf{0}, \beta^T \geq \mathbf{0}$’ implies ‘$\mathbf{x}^T\mathbf{d} \leq \mathbf{0}$’;
i.e., iff ‘$\mathbf{A}\mathbf{x} \leq \mathbf{0}$’ implies ‘$\mathbf{d}^T\mathbf{x} \leq \mathbf{0}$.’

□

The analogue of ‘vector spaces’ for inequality systems is ‘**cones**’. A **cone** is a collection of vectors closed under addition and non-negative linear combination. It is easily verified that the solution set of $\mathbf{A}\mathbf{x} \geq \mathbf{0}$ is a cone. Such cones are called **polyhedral**. We say vectors $\mathbf{x}$, $\mathbf{y}$ (on the same set S) are **polar** iff $< \mathbf{x}, \mathbf{y} >$ (i.e., their dot product) is nonpositive. If $\mathcal{K}$ is a collection of vectors, the **polar of** $\mathcal{K}$, denoted by $\mathcal{K}^p$ is the collection of vectors polar to every vector in $\mathcal{K}$. Thus, Farkas Lemma states:
‘Let $\mathcal{C}$ be the polyhedral cone defined by $\mathbf{A}\mathbf{x} \leq \mathbf{0}$. A vector $\mathbf{d}$ belongs to $\mathcal{C}^p$ iff $\mathbf{d}^T$ is a nonnegative linear combination of the rows of $\mathbf{A}$.’

1.3.2 Linear Programming

Let $\mathcal{S}$ be a linear inequality system with ‘$\leq$’ and ‘$=$’ constraints (‘$\geq$’ and ‘$=$’ constraints). The **linear programming problem** or **linear program** is to finda solution $\mathbf{x}$ of $\mathcal{S}$ which maximizes a given linear function $\mathbf{c}^T\mathbf{x}$ (minimizes a given linear function $\mathbf{c}^T\mathbf{x}$). The linear function to be optimized is called the **objective function**. A solution of $\mathcal{S}$ is called a **feasible solution**, while a solution which optimizes $\mathbf{c}^T\mathbf{x}$ is called an **optimal solution**, of the linear programming problem.

The **value** of a feasible solution is the value of the objective function on it.

The following linear programming problems are said to be **duals** of each other

Primal program

$$\text{Maximize} \qquad \mathbf{c}_1^T\mathbf{x}_1 + \mathbf{c}_2^T\mathbf{x}_2$$

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \end{pmatrix} \begin{matrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{matrix} = \mathbf{b}_1$$

$$\begin{pmatrix} \mathbf{A}_{21} & \mathbf{A}_{22} \end{pmatrix} \begin{matrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{matrix} \leq \mathbf{b}_2$$

$$\mathbf{x}_2 \geq 0$$

Dual program

$$\text{Minimize} \qquad \mathbf{b}_1^T\mathbf{y}_1 + \mathbf{b}_2^T\mathbf{y}_2$$

$$\begin{pmatrix} \mathbf{A}_{11}^T & \mathbf{A}_{21}^T \end{pmatrix} \begin{matrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{matrix} = \mathbf{c}_1$$

$$\begin{pmatrix} \mathbf{A}_{12}^T & \mathbf{A}_{22}^T \end{pmatrix} \begin{matrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{matrix} \geq \mathbf{c}_2$$

$$\mathbf{y}_2 \geq 0.$$

We now present the duality theorem of linear programming [von Neumann47], [Gale+Kuhn+Tucker51].

Theorem 1.3.3 *For dual pairs of linear programs the following statements hold:*

i. The value of each feasible solution of the minimization program is greater than or equal to the value of each feasible solution of the maximization program;

ii. if both programs have feasible solutions then both have optimal solutions and the optimal values are equal;

iii. if one program has an optimal solution then so does the other.

The usual proof uses Farkas Lemma, or more conveniently, Kuhn-Fourier Theorem. We only sketch it.
Sketch of Proof: Part (i) follows by the solution of Exercise 1.1. Now we write down the inequalities of the primal and dual programs and another '$\leq$' inequality which is the **opposite** of the inequality in part (i). Part (ii) would be proved if this system of inequalities has a solution. We assume it has no solution and derive a contradiction by using Kuhn-Fourier Theorem.

$\square$

Exercise 1.1 *Prove part (i) of Theorem 1.3.3.*

A very useful corollary of Theorem 1.3.3 is the following:

Corollary 1.3.1 *(***Complementary Slackness***)*
Let

$$\left\{ \begin{array}{c} \max \mathbf{c}^T\mathbf{x} \\ \mathbf{A}\mathbf{x} = \mathbf{b} \\ \mathbf{x} \geq \mathbf{0} \end{array} \right\} \textit{ and } \left\{ \begin{array}{c} \min \mathbf{b}^T\mathbf{y} \\ \mathbf{A}^T\mathbf{y} \geq \mathbf{c} \end{array} \right\}$$

be dual linear programs. Let $\hat{\mathbf{x}}, \hat{\mathbf{y}}$ be optimal solutions to the respective programs. Then,

i. $\hat{x}_i > 0$ implies $(\mathbf{A}^T)_i\hat{\mathbf{y}} = c_i$,

ii. $(\mathbf{A}^T)_i\hat{\mathbf{y}} > c_i$ implies $\hat{\mathbf{x}}_i = 0$.

Proof : We have by part (ii) of Theorem 1.3.3 $\mathbf{c}^T\hat{\mathbf{x}} = \hat{\mathbf{y}}^T\mathbf{b}$, equivalently

$$\mathbf{c}^T\hat{\mathbf{x}} - \hat{\mathbf{y}}^T\mathbf{A}\hat{\mathbf{x}} = (\mathbf{c}^T - \hat{\mathbf{y}}^T\mathbf{A})\hat{\mathbf{x}} = 0.$$

The result now follows since $(\mathbf{c}^T - \hat{\mathbf{y}}^T\mathbf{A}) \geq 0$ and $\hat{\mathbf{x}} \geq \mathbf{0}$.

$\square$

1.4 Solutions of Exercises

E 1.1: We use the linear programs given in the definition of dual linear programs. We have

$$\begin{aligned} \begin{pmatrix} \mathbf{b}_1^T & \mathbf{b}_2^T \end{pmatrix} \begin{matrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{matrix} &\geq \left(\begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{pmatrix} \begin{bmatrix} \mathbf{A}_{11} & \mathbf{A}_{12} \\ \mathbf{A}_{21} & \mathbf{A}_{22} \end{bmatrix}^T \right) \begin{matrix} \mathbf{y}_1 \\ \mathbf{y}_2 \end{matrix} \\ &\geq \begin{pmatrix} \mathbf{x}_1^T & \mathbf{x}_2^T \end{pmatrix} \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}. \end{aligned}$$

1.5 Solutions of Problems

Most of these problems can be found as standard results in undergraduate texts on linear algebra (see for instance [Hoffman+Kunze72]). We only give the solution to the last two problems. Here we follow [MacDuffee33], [Gantmacher59] respectively.

P 1.16: We state the following simple lemma without proof.

Lemma 1.5.1 *If $\alpha_1, \cdots, \alpha_t$ are permutations of $\{1, \cdots, n\}$ then $sgn(\alpha_1\alpha_2\cdots\alpha_t) = (sgn(\alpha_1))(sgn(\alpha_2))\cdots(sgn(\alpha_t))$ (where $\alpha_i\alpha_j$ denotes composition of permutations α_i, α_j).*

We have

$$\sum sgn(\sigma)\ det\left(\mathbf{A}\begin{pmatrix} r_1, & \cdots & , r_k \\ i_1, & \cdots & , i_k \end{pmatrix}\right) det\left(\mathbf{A}\begin{pmatrix} r_{k+1}, & \cdots & , r_m \\ i_{k+1}, & \cdots & , i_m \end{pmatrix}\right) =$$
$$\sum sgn(\sigma)(\sum sgn(\alpha)(a_{r_1\alpha(i_1)}\cdots a_{r_k\alpha(i_k)}))(\sum sgn(\beta)(a_{r_{k+1}\beta(i_{k+1})}\cdots a_{r_n\beta(i_n)})),$$

where α, β are permutations on the sets $\{i_1, \cdots, i_k\}$, $\{i_{k+1}, \cdots, i_n\}$ respectively. Let α' agree with α over $\{i_1, \cdots, i_k\}$ and over $\{i_{k+1}, \cdots, i_n\}$, with the identity permutation. Let β' agree with β over $\{i_{k+1}, \cdots, i_n\}$ and with the identity permutation over $\{i_1, \cdots, i_k\}$. So

$$\begin{aligned} LHS &= \sum sgn(\sigma)sgn(\alpha')sgn(\beta')(a_{r_1\alpha(i_1)}\cdots a_{r_k\alpha(i_k)}a_{r_{k+1}\beta(i_{k+1})}\cdots a_{r_n\beta(i_n)}) \\ &= \sum sgn(\beta'\alpha'\sigma)(a_{r_1\alpha\sigma(r_1)}\cdots a_{r_k\alpha\sigma(r_k)}a_{r_{k+1}\beta\sigma(r_{k+1})}\cdots a_{r_n\beta\sigma(r_n)}) \\ &= \sum sgn(\mu)(a_{r_1\mu(r_1)}\cdots a_{r_k\mu(r_k)}a_{r_{k+1}\mu(r_{k+1})}\cdots a_{r_n\mu(r_n)}), \end{aligned}$$

where $\mu \equiv \beta'\alpha'\sigma$. Since the RHS is the usual definition of the determinant of $\mathbf{A}$, the proof is complete.

P 1.17: Let a_{ij}, b_{ij} denote respectively the $(i,j)^{th}$ entry of $\mathbf{A}, \mathbf{B}$. Then the matrix

$$\mathbf{AB} = \begin{bmatrix} \sum_{i_1=1}^n a_{1i_1} b_{i_1 1} & \cdots & \sum_{i_m=1}^n a_{1i_m} b_{i_m m} \\ \vdots & & \vdots \\ \sum_{i_1=1}^n a_{mi_1} b_{i_1 1} & \cdots & \sum_{i_m=1}^n a_{mi_m} b_{i_m m} \end{bmatrix}.$$

Now each column of $\mathbf{AB}$ can be thought of as the sum of n appropriate columns - for instance the transpose of the first column is made up of rows - a typical one being $(a_{1i_1} b_{i_1 1}, \cdots, a_{mi_1} b_{i_1 1})$. Using Theorem 1.2.8

$$\begin{aligned} det(\mathbf{AB}) &= \sum_{i_1,\cdots,i_m} det\left(\begin{bmatrix} a_{1i_1} b_{i_1 1} & \cdots & a_{1i_m} b_{i_m m} \\ \vdots & & \vdots \\ a_{mi_1} b_{i_1 1} & \cdots & a_{mi_m} b_{i_m m} \end{bmatrix}\right) \\ &= \sum (b_{i_1 1} \cdots b_{i_m m}) \;\; det\left(\mathbf{A}\begin{pmatrix} 1, & \cdots & , m \\ i_1, & \cdots & , i_m \end{pmatrix}\right), \end{aligned}$$

where $\mathbf{A}\begin{pmatrix} 1, & \cdots & , m \\ i_1, & \cdots & , i_m \end{pmatrix}$ is the $m \times m$ matrix which has the first m rows of $\mathbf{A}$ in the same order as in $\mathbf{A}$ but whose j^{th} column is the i_j^{th} column of $\mathbf{A}$. So, again by Theorem 1.2.8,

$$\det(\mathbf{AB}) = \sum_{k_1,\cdots,k_m} det\left(\mathbf{A}\begin{pmatrix} 1, & \cdots & , m \\ k_1, & \cdots & , k_m \end{pmatrix}\right) (sgn(\sigma)) \; b_{\sigma(k_1)1} \cdots b_{\sigma(k_m)m},$$

where $k_1 < \cdots < k_m, \{k_1, \cdots, k_m\} = \{i_1, \cdots, i_m\}$ and σ is the permutation
$\begin{pmatrix} k_1, & \cdots & , k_m \\ i_1, & \cdots & , i_m \end{pmatrix}$, i.e.,

$$\sigma(k_j) = i_j.$$

So,

$$det(\mathbf{AB}) = \sum_{\substack{k_1, \cdots, k_m \\ k_1 < \cdots < k_m}} det\left(\mathbf{A}\begin{pmatrix} 1, & \cdots & , m \\ k_1, & \cdots & , k_m \end{pmatrix}\right) det\left(\mathbf{B}\begin{pmatrix} k_1, & \cdots & , k_m \\ 1, & \cdots & , m \end{pmatrix}\right).$$

Chapter 2

Graphs

2.1 Introduction

We give definitions of graphs and related notions below. Graphs should be visualized as points joined by lines with or without arrows rather than be thought of as formal objects. We would not hesitate to use informal language in proofs.

2.2 Graphs: Basic Notions

2.2.1 Graphs and Subgraphs

A graph $\mathcal{G}$ is a triple $(V(\mathcal{G}), E(\mathcal{G}), i_{\mathcal{G}})$ where $V(\mathcal{G})$ is a finite set of **vertices**, $E(\mathcal{G})$ is a finite set of **edges** and $i_{\mathcal{G}}$ is an **incidence function** which associates with each edge a pair of vertices, not necessarily distinct, called its **end points** or **end vertices** (i.e., $i_{\mathcal{G}} : E(\mathcal{G}) \rightarrow$ collection of subsets of $V(\mathcal{G})$ of cardinality 2 or 1).
Vertices are also referred to as **nodes** or **junctions** while edges are referred to also as **arcs** or **branches**.
We note

i. an edge may have a single end point - such edges are called **selfloops**.

ii. a vertex may have no edges incident on it - such vertices are said to be **isolated**.

iii. the graph may be in several pieces.

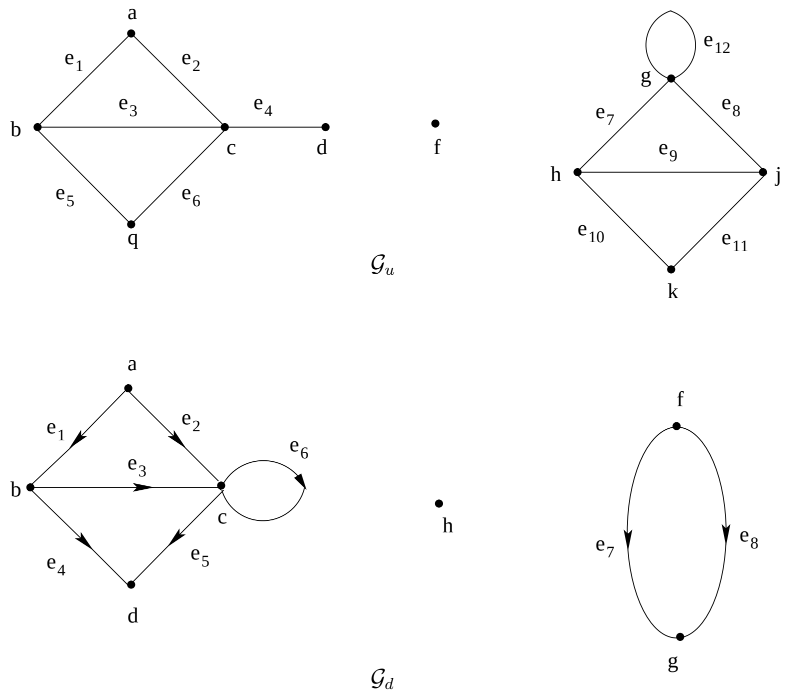


Figure 2.1: Undirected and Directed Graphs

Figure 2.1 shows a typical graph $\mathcal{G}_u$.

A **directed** graph $\mathcal{G}$ is a triple $(V(\mathcal{G}), E(\mathcal{G}), a_{\mathcal{G}})$ where $V(\mathcal{G}), E(\mathcal{G})$ are the vertex set and the edge set respectively and $a_{\mathcal{G}}$ associates with each edge an ordered pair of vertices not necessarily distinct (i.e., $a_{\mathcal{G}} : E(\mathcal{G}) \rightarrow V(\mathcal{G}) \times V(\mathcal{G})$). The first element of the ordered pair is the **positive end point** or **tail** of the arrow and the second element is the **negative end point** or **head** of the arrow. For selfloops, positive and negative endpoints are the same. Directed graphs are usually drawn as graphs with arrows in the edges. In Figure 2.1, $\mathcal{G}_d$ is a directed graph.
We say a vertex v and an edge e are **incident** on each other iff v is an end point of e. If e has end points u, v we say that u, v are **adjacent**

to each other. Two edges e_1, e_2 are **adjacent** if they have a common end point. The **degree** of a vertex is the number of edges incident on it with selfloops counted twice.
A graph $\mathcal{G}_s$ is a **subgraph** of $\mathcal{G}$ iff $\mathcal{G}_s$ is a graph, $V(\mathcal{G}_s) \subseteq V(\mathcal{G})$, $E(\mathcal{G}_s) \subseteq E(\mathcal{G})$, and the endpoints of an edge in $\mathcal{G}_s$ are the same as its end points in $\mathcal{G}$.
Subgraph $\mathcal{G}_s$ is a **proper subgraph** of $\mathcal{G}$ iff it is a subgraph of $\mathcal{G}$ but not identical to it. The **subgraph** of $\mathcal{G}$ on V_1 is that subgraph of $\mathcal{G}$ which has V_1 as its vertex set and the set of edges of $\mathcal{G}$ with both end points in V_1 as the edge set. The **subgraph** of $\mathcal{G}$ on E_1 has $E_1 \subseteq E(\mathcal{G})$ as the edge set and the endpoints of edges in E_1 as the vertex set. If $\mathcal{G}$ is a directed graph the edges of a subgraph would retain the directions they had in $\mathcal{G}$ (i.e., they would have positive and negative end points as in $\mathcal{G}$).

Exercise 2.1 *(k) In any graph with atleast two nodes and no parallel edges (edges with the same end points) or selfloops show that the degree of some two vertices must be equal.*

Exercise 2.2 *(k) Show that*

i. the sum of the degrees of vertices of any graph is equal to twice the number of edges of the graph;

ii. the number of odd degree vertices in any graph must be even.

2.2.2 Connectedness

A **vertex edge alternating sequence** (**alternating sequence** for short) of a graph $\mathcal{G}$ is a sequence in which

i. vertices and edges of $\mathcal{G}$ alternate,

ii. the first and last elements are vertices and

iii. whenever a vertex and an edge occur as adjacent elements they are incident on each other in the graph.

Example: For the graph $\mathcal{G}_u$ in Figure 2.1, $(a, e_1, b, e_3, c, e_6, q, e_6, c, e_4, d)$ is an alternating sequence.
A **path** is a graph all of whose edges and vertices can be arranged in an alternating sequence without repetitions.
It can be seen that the degree of precisely two of the vertices of the path is one and the degree of all other vertices (if any) is two. The two vertices of degree one must appear at either end of any alternating sequence containing all nodes and edges of the path without repetition. They are called **terminal nodes**. The path is said to be **between** its terminal nodes. It is clear that there are only two such alternating sequences that we can associate with a path. Each is the reverse of the other. The two alternating sequences associated with the path in Figure 2.2 are $(v_1, e_1, v_2, e_2, v_3, e_3, v_4)$ and $(v_4, e_3, v_3, e_2, v_2, e_1, v_1)$.

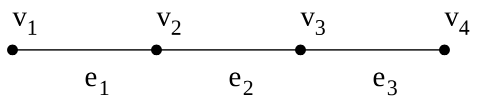


Figure 2.2: A Path Graph

We say 'go along the path from v_i to v_j' instead of 'construct the alternating sequence without repetitions having v_i as the first element and v_j as the last element'. Such sequences are constructed by considering the alternating sequence associated with the path in which v_i precedes v_j and taking the subsequence starting with v_i and ending with v_j.
A directed graph may be a path if it satisfies the above conditions. However, the term **strongly directed path** is used if the edges can be arranged in a sequence so that the negative end point of each edge, except the last is the positive end point of the succeeding edge.
A graph is said to be **connected** iff for any given pair of distinct vertices there exists a path subgraph between them. The path graph in Figure 2.2 is connected while the graph $\mathcal{G}_u$ in Figure 2.1 is disconnected.
A **connected component** of a graph $\mathcal{G}$ is a connected subgraph of $\mathcal{G}$ that is not a proper subgraph of any connected subgraph of $\mathcal{G}$ (i.e., it is a maximal connected subgraph). Connected components correspond to 'pieces' of a disconnected graph.

Exercise 2.3 *(k) Let $\mathcal{G}$ be a connected graph. Show that there is a vertex such that if the vertex and all edges incident on it are removed the remaining graph is still connected.*

2.2.3 Circuits and Cutsets

A connected graph with each vertex having degree two is called a **circuit graph** or a **polygon graph**. ($\mathcal{G}_L$ in Figure 2.3 is a circuit graph). If $\mathcal{G}'$ is a circuit subgraph of $\mathcal{G}$ then $E(\mathcal{G}')$ is a **circuit** of $\mathcal{G}$. A single edged circuit is called a **selfloop**.

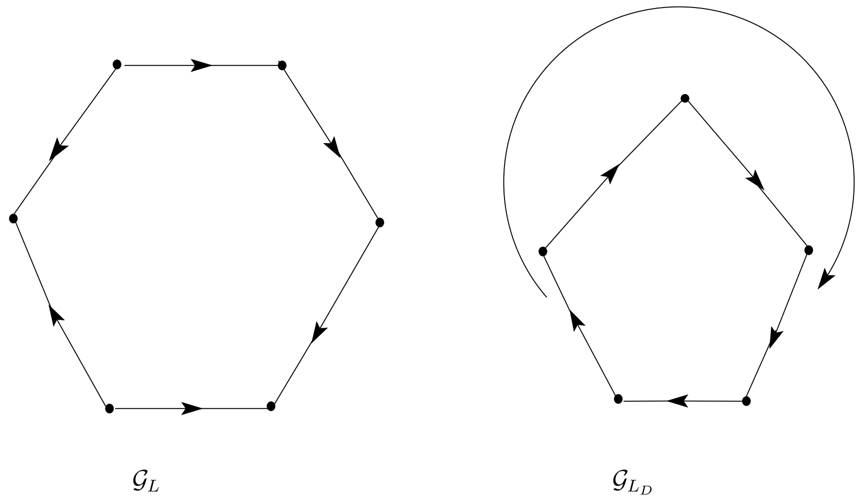


Figure 2.3: A Circuit Graph and a Strongly Directed Circuit Graph

Each of the following is a characteristic property of circuit graphs (i.e., each can be used to define the notion).
We omit the routine proofs.

i. A circuit graph has precisely two paths between any two of its vertices.

ii. If we start from any vertex v of a circuit graph and follow any path (i.e., follow an edge, reach an adjacent vertex, go along a new edge incident on that vertex and so on) the first vertex to be repeated would be v. Also during the traversal we would have encountered all vertices and edges of the circuit graph.

iii. Deletion of any edge (leaving the end points in place) of a circuit graph reduces it to a path.

Exercise 2.4 *Construct*

i. a graph with all vertices of degree 2 *that is not a circuit graph,*

ii. a non circuit graph which is made up of a path and an additional edge,

iii. a graph which has no circuits,

iv. a graph which has every edge as a circuit.

Exercise 2.5 *Prove*

Lemma 2.2.1 *(k) Deletion of an edge (leaving end points in place) of a circuit subgraph does not increase the number of connected components in the graph.*

Exercise 2.6 *Prove*

Theorem 2.2.1 *(k) A graph contains a circuit if it contains two distinct paths between some two of its vertices.*

Exercise 2.7 *Prove*

Theorem 2.2.2 *(k) A graph contains a circuit if every one of its vertices has degree* ≥ 2.

A set $T \subseteq E(\mathcal{G})$ is a **crossing edge set** of $\mathcal{G}$ if $V(\mathcal{G})$ can be partitioned into sets V_1, V_2 such that $T = \{e : e$ has an end point in V_1 and in $V_2\}$. (In Figure 2.4, C is a crossing edge set). We will call V_1, V_2 the **end vertex sets** of T. Observe that while end vertex sets uniquely determine a crossing edge set there may be more than one pair of end vertex sets consistent with a given crossing edge set. A crossing edge set that is minimal (i.e., does not properly contain another crossing edge set) is called a **cutset** or a **bond**. A single edged cutset is a **coloop**.

Exercise 2.8 *Construct a graph which has (a) no cutsets (b) every edge as a cutset.*

Exercise 2.9 *Construct a crossing edge set that is not a cutset (see Figure 2.4).*

Exercise 2.10 *(k) Show that a cutset is a minimal set of edges with the property that when it is deleted leaving endpoints in place the number of components of the graph increases.*

Exercise 2.11 *Short (i.e., fuse end points of an edge and remove the edge) all branches of a graph except a cutset. How does the resulting graph look?*

Exercise 2.12 *Prove*

Theorem 2.2.3 *(k) A crossing edge set T is a cutset iff it satisfies the following:*

i. If the graph has more than one component then T must meet the edges of only one component and

ii. if the end vertex sets of T are V_1, V_2 in that component, then the subgraphs on V_1 and V_2 must be connected.

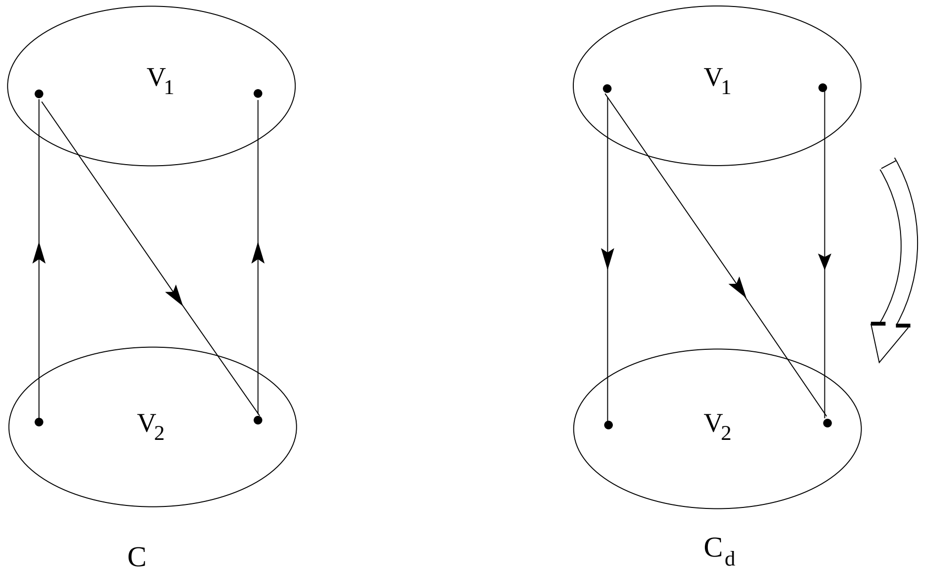


Figure 2.4: A Crossing Edge Set and a Strongly Directed Crossing Edge Set

2.2.4 Trees and Forests

A graph that contains no circuits is called a **forest graph** (see graphs $\mathcal{G}_t$ and $\mathcal{G}_f$ in Figure 2.5). A connected forest graph is also called a **tree graph** (see graph $\mathcal{G}_t$ in Figure 2.5).

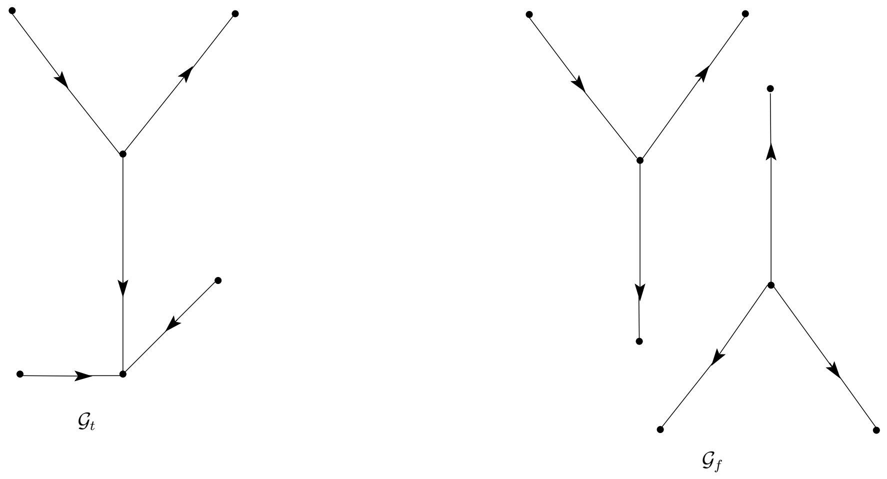


Figure 2.5: A Tree Graph $\mathcal{G}_t$ and a Forest Graph $\mathcal{G}_f$

A **forest** of a graph $\mathcal{G}$ is the set of edges of a forest subgraph of $\mathcal{G}$ that has $V(\mathcal{G})$ as its vertex set and has as many connected components as $\mathcal{G}$ has. A forest of a connected graph $\mathcal{G}$ is also called a **tree** of $\mathcal{G}$. The complement relative to $E(\mathcal{G})$ of a forest (tree) is a **coforest (cotree)** of $\mathcal{G}$. The number of edges in a forest (coforest) of $\mathcal{G}$ is its **rank (nullity)**. Theorem 2.2.4 assures us that this notion is well defined.

Exercise 2.13 *(k) Show that a tree graph on two or more nodes has*

i. precisely one path between any two of its vertices

ii. at least two vertices of degree one.

Exercise 2.14 *Prove*

Theorem 2.2.4 *(k) A tree graph on n nodes has $(n-1)$ branches. Any connected graph on n nodes with $(n-1)$ edges is a tree graph.*

Corollary 2.2.1 *The forest subgraph on n nodes and p components has $(n-p)$ edges.*

Exercise 2.15 *Prove*

Theorem 2.2.5 *(k) A subset of edges of a graph is a forest (coforest) iff it is a maximal subset not containing any circuit (cutset).*

Exercise 2.16 *(k) Show that every forest (coforest) of a graph $\mathcal{G}$ intersects every cutset (circuit) of $\mathcal{G}$.*

Exercise 2.17 *Prove*

Lemma 2.2.2 *(k) A tree graph splits into two tree graphs if an edge is opened (deleted leaving its end points in place).*

Exercise 2.18 *(k) Show that a tree graph yields another tree graph if an edge is shorted (removed after fusing its end points).*

Exercise 2.19 *Prove*

Theorem 2.2.6 *(k) Let f be a forest of a graph $\mathcal{G}$ and let e be an edge of $\mathcal{G}$ outside f. Then $e \cup f$ contains only one circuit of $\mathcal{G}$.*

Exercise 2.20 *Prove*

Theorem 2.2.7 *(k) Let $\overline{f}$ be a coforest of a graph $\mathcal{G}$ and let e be an edge of $\mathcal{G}$ outside $\overline{f}$ (i.e., $e \in f$). Then $e \cup \overline{f}$ contains only one cutset of $\mathcal{G}$ (i.e., only one cutset of $\mathcal{G}$ intersects f in e).*

Exercise 2.21 *(k) Show that every circuit is an f-circuit with respect to some forest (i.e., intersects some coforest in a single edge).*

Exercise 2.22 *(k) Show that every cutset is an f-cutset with respect to some forest (i.e., intersects some forest in a single edge).*

Exercise 2.23 *(k) Show that shorting an edge in a cutset of a graph does not reduce the nullity of the graph.*

2.2.5 Strongly Directedness

The definitions we have used thus far hold also in the case of directed graphs. The subgraphs in each case retain the original orientation for the edges. However, the prefix 'strongly directed' in each case implies a stronger condition. We have already spoken of the strongly directed path. A strongly directed circuit graph has its edges arranged in a sequence so that the negative end point of each edge is the positive end point of the succeeding edge and the positive end point of the last edge is the negative end point of the first (see $\mathcal{G}_{L_d}$ in Figure 2.3). The set of edges of such a graph would be a **strongly directed circuit**.

A **strongly directed crossing edge set** would have the positive end points of all its edges set in the same end vertex set (see C_d in Figure 2.4).

In this book we will invariably assume that the graph is directed but our circuit subgraphs, paths etc. although they are directed graphs, will, unless otherwise stated, not be strongly directed. When it is clear from the context the prefix 'directed' will be omitted when we speak of a graph. For simplicity we would write directed path, directed circuit, directed crossing edge set instead of strongly directed path etc.

Exercise 2.24 *Prove:*
(Minty) Any edge of a directed graph is either in a directed circuit or in a directed cutset but not both.

(For solution see Theorem 2.4.7).

2.2.6 Fundamental Circuits and Cutsets

Let f be a forest of $\mathcal{G}$ and let $e \notin f$. It can be shown (Theorem 2.2.6) that there is a unique circuit contained in $e \cup f$. This circuit is called the **fundamental circuit (f - circuit) of e with respect to f** and is denoted by $L(e, f)$. Let $e_t \in f$. It can be shown (Theorem 2.2.7) that there is a unique cutset contained in $e_t \cup \bar{f}$. This cutset is called the **fundamental cutset of** e_t **with respect to f** and is denoted by $B(e_t, f)$.

Remark: The f-circuit $L(e, f)$ is obtained by adding e to the unique path in the forest subgraph on f between the end points of e. For the subgraph on f, the edge e_t is a crossing edge set with end vertex sets say V_1, V_2. Then the f-cutset $B(e_t, f)$ is the crossing edge set of $\mathcal{G}$ with end vertex sets V_1, V_2.

2.2.7 Orientation

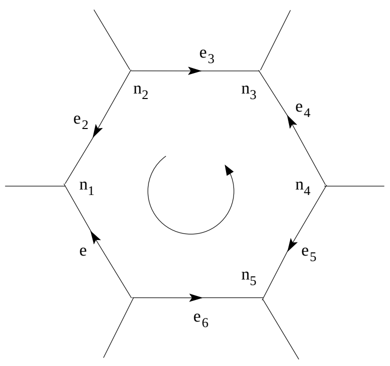
Let $\mathcal{G}$ be a directed graph. We associate **orientations** with circuit subgraphs and crossing edge sets as follows:

An **orientation** of a circuit subgraph is an alternating sequence of its vertices and edges, without repetitions except for the first vertex being also the last (note that each edge is incident on the preceding and succeeding vertices). Two orientations are **equivalent** if one can be obtained by a **cyclic** shift of the other. Diagrammatically an orientation may be represented by a circular arrow. It is easily seen that there can be at most two orientations for a circuit graph. (A single edge circuit subgraph has only one). These are obtained from each other by reversing the sequence. When there are two non equivalent orientations we call them **opposite** to each other. We say that an edge of the circuit subgraph **agrees** with the **orientation** if its positive end point immediately precedes itself in the orientation (or in an equivalent orientation). Otherwise it is opposite to the orientation.
The orientation associated with a circuit subgraph would also be called the **orientation** of the **circuit**.

Example: For the circuit subgraph of Figure 2.6 the orientations $(n_1, e,\ n_6,\ e_6,\ n_5,\ e_5,\ n_4,\ e_4,\ n_3,\ e_3,\ n_2,\ e_2,\ n_1)$, and $(n_6,\ e_6,\ n_5,\ e_5,$ $n_4,\ e_4,\ n_3,\ e_3,\ n_2,\ e_2,\ n_1, e, n_6)$ are equivalent. This is the orientation shown in the figure. It is opposite to the orientation $(n_1, e_2,\ n_2,\ e_3,$ $n_3,\ e_4,\ n_4,\ e_5,\ n_5,\ e_6,\ n_6, e, n_1)$. The edge e **agrees** with this latter orientation and is **opposite** to the former orientation.

An **orientation** of a crossing edge set is an ordering of its end vertex sets V_1, V_2 as (V_1, V_2) or as (V_2, V_1). An edge e in the crossing edge set with positive end point in V_1 and negative end point in V_2 **agrees** with the orientation (V_1, V_2) and is **opposite** to the orientation (V_2, V_1). In Figure 2.6 the orientation of the crossing edge set is (V_1, V_2).

Theorem 2.2.8 *(k) Let f be a forest of a directed graph $\mathcal{G}$. Let $e_t \in f$ and let $e_c \in \overline{f}$. Let the orientation of $L(e_c, f)$ and $B(e_t, f)$ agree with e_c, e_t, respectively. Then $L(e_c, f) \cap B(e_t, f) = \emptyset$ or $\{e_c, e_t\}$.*
Further when the intersection is nonvoid e_t agrees with (opposes) the orientation of $L(e_c, f)$ iff e_c opposes (agrees with) the orientation of $B(e_t, f)$.



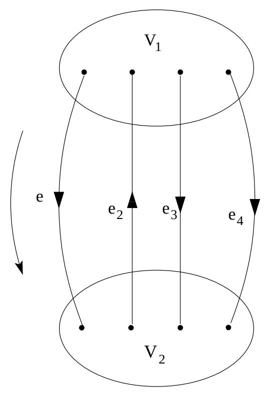


Figure 2.6: Circuit subgraph and Crossing Edge Set with Orientations

Proof : We confine ourselves to the case where $\mathcal{G}$ is connected since even if it is disconnected we could concentrate on the component where e_t is present.
If $B(e_t, f)$ is deleted from $\mathcal{G}$, two connected subgraphs $\mathcal{G}_1, \mathcal{G}_2$ result whose vertex sets are the end vertex sets V_1, V_2, respectively of $B(e_t, f)$. Now e_c could have both end points in V_1, both end points in V_2, or one end point in V_1 and another in V_2. In the former two cases $L(e_c, f) \cap B(e_t, f) = \emptyset$. In the last case $L(e_c, f)$ must contain e_t. For, the path in the tree subgraph on f between the endpoints of e_c must use e_t since that is the only edge in f with one endpoint in V_1 and the other in V_2. Now $L(e_c, f)$ contains only one edge, namely e_c from $\overline{f}$ and $B(e_t, f)$ contains only one edge, namely e_t from f. Hence in the third case

$$L(e_c, f) \cap B(e_t, f) = \{e_c, e_t\}.$$

Let us next assume that the intersection is nonvoid. Suppose that e_c has its positive end point a in V_1 and negative end point b in V_2. Let $(b, \cdots, e_t, \cdots, a, e_c, b)$ be an orientation of the circuit. It is clear that e_t would agree with this orientation if V_2 contains its positive end point and V_1 its negative end point (see Figure 2.7). But in that case e_c would oppose the orientation of $B(e_t, f)$ (which is (V_2, V_1), taken to agree with the orientation of e_t). The other cases can be handled similarly.

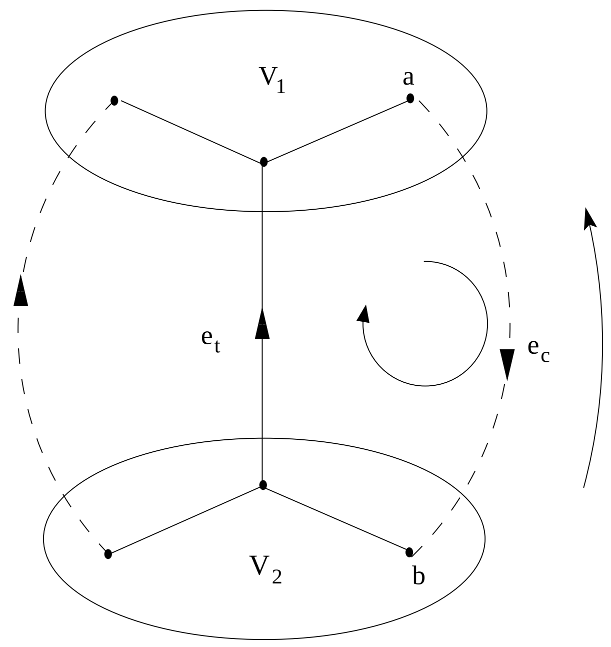


Figure 2.7: Relation between f-circuit and f-cutset

2.2.8 Isomorphism

Let $\mathcal{G}_1 \equiv (V_1, E_1, i_1)$, $\mathcal{G}_2 \equiv (V_2, E_2, i_2)$, be two graphs. We say $\mathcal{G}_1$, $\mathcal{G}_2$ are **identical** iff $V_1 = V_2, E_1 = E_2$ and $i_1 = i_2$. However, graphs could be treated as essentially the same even if they satisfy weaker conditions. We say $\mathcal{G}_1$, $\mathcal{G}_2$ are **isomorphic** to each other and denote it by (abusing notation) $\mathcal{G}_1 = \mathcal{G}_2$ iff there exist bijections $\eta : V_1 \rightarrow V_2$ and $\epsilon : E_1 \rightarrow E_2$ s.t. any edge e has end points a, b in $\mathcal{G}_1$ iff $\epsilon(e)$ has endpoints $\eta(a), \eta(b)$. If $\mathcal{G}_1, \mathcal{G}_2$ are directed graphs then we would further require that an end point a of e, in $\mathcal{G}_1$, is positive (negative) iff $\eta(a)$ is the positive (negative) endpoint of $\epsilon(e)$. When we write $\mathcal{G}_1 = \mathcal{G}_2$ usually the bijections would be clear from the context. However, when two graphs are isomorphic there would be many **isomorphisms** ((η, ϵ) pairs) between them.

The graphs $\mathcal{G}, \mathcal{G}'$ in Figure 2.8 are isomorphic. The node and edge bijections are specified by the ('). Clearly there is at least one other (η, ϵ) pair between the graphs.

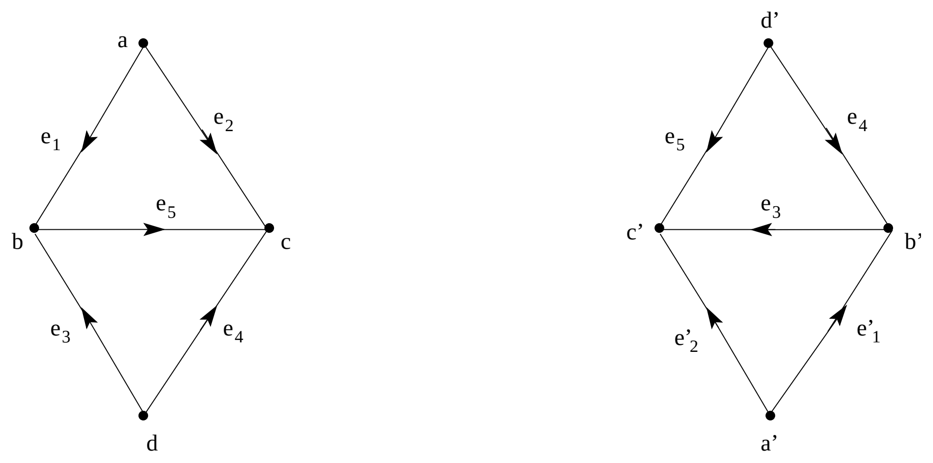


Figure 2.8: Isomorphic Directed Graphs

2.2.9 Cyclically connectedness

A graph $\mathcal{G}$ is said to be **cyclically connected** iff given any pair of vertices there is a circuit subgraph containing them.

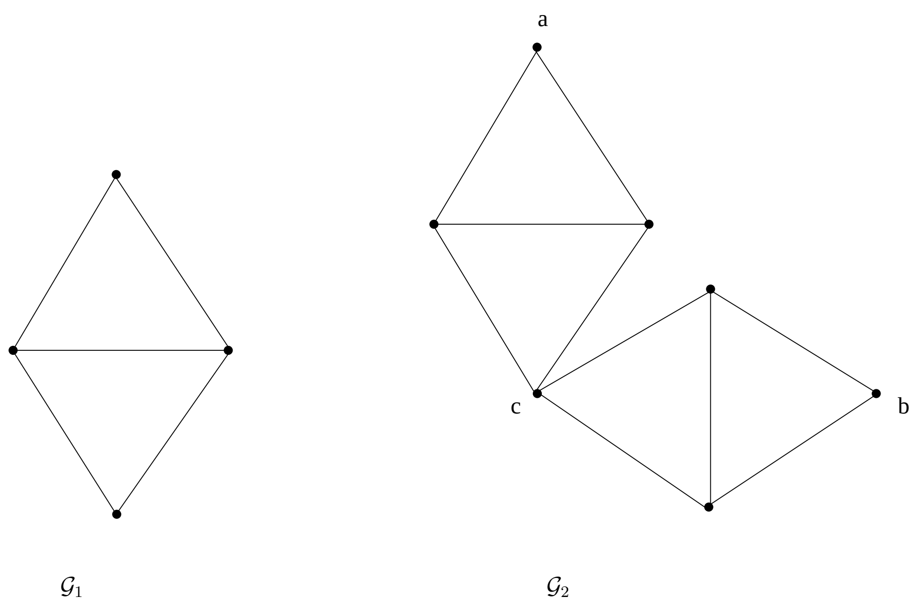


Figure 2.9: Cyclically Connected and Cyclically Disconnected Graphs

Example: The graph $\mathcal{G}_1$ in Figure 2.9 is cyclically connected while $\mathcal{G}_2$ of the same figure is not cyclically connected since no circuit subgraph contains both nodes a and b.
Whenever a connected graph is not cyclically connected there would be two vertices a, b through which no circuit subgraph passes. If a, b are not joined by an edge there would be a vertex c such that every path between a and b passes through c. We then say c is a **cut vertex** or **hinge**. The graph $\mathcal{G}_2$ of Figure 2.9 has c as a cut vertex.
It can be shown that a graph is cyclically connected iff any pair of edges can be included in the same circuit.

In any graph it can be shown that if edges e_1, e_2 and e_2, e_3 belong to circuits C_{12}, C_{23}, then there exists a circuit $C_{13} \subseteq C_{12} \cup C_{23}$ s.t. $e_1, e_3 \in C_{13}$. It follows that the edges of a graph can be partitioned into blocks such that within each block every pair of distinct edges can be included in some circuit and edges belonging to different blocks cannot be included in the same circuit (each coloop would form a block by itself). We will call such a block an **elementary separator** of the graph. Unions of such blocks will be called **separators**. The subgraphs on elementary separators will be called **2-connected components**.(Note that a coloop is a 2-connected component by itself). If two 2-connected components intersect they would do so at a single vertex which would be a cut vertex. If two graphs have a single common vertex, we would say that they are put together by **hinging**.

2.3 Graphs and Vector Spaces

There are several natural 'electrical' vectors that one may associate with the vertex and edge sets of a directed graph $\mathcal{G}$.

e.g.
- i. potential vectors on the vertex set,
- ii. current vectors on the edge set,
- iii. voltage (potential difference) vectors on the edge set.

Our concern will be with the latter two examples. We need a few preliminary definitions. Henceforth, unless otherwise specified, by **graph** we mean **directed graph**.

The Incidence Matrix

The **incidence matrix A** of a graph $\mathcal{G}$ is defined as follows:
A has one row for each node and one column for each edge.

$$\mathbf{A}(i,j) \quad = \quad \begin{array}{l} +1(-1) \text{ if edge } j \text{ has its arrow leaving (entering) node } i. \\ 0 \text{ if edge } j \text{ is not incident on node } i \\ \quad \text{or if edge } j \text{ is a selfloop.} \end{array}$$

Example: The incidence matrix of the directed graph $\mathcal{G}_d$ in Figure 2.1 is

$$\mathbf{A} = \begin{array}{c|cccccc|cc} & e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 & e_8 \\ a & +1 & +1 & 0 & 0 & 0 & 0 & 0 & 0 \\ b & -1 & 0 & +1 & +1 & 0 & 0 & 0 & 0 \\ c & 0 & -1 & -1 & 0 & +1 & 0 & 0 & 0 \\ d & 0 & 0 & 0 & -1 & -1 & 0 & 0 & 0 \\ \hline f & 0 & 0 & 0 & 0 & 0 & 0 & +1 & +1 \\ g & 0 & 0 & 0 & 0 & 0 & 0 & -1 & -1 \\ \hline h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{array} \tag{2.1}$$

Note that the selfloop e_6 is represented by a zero column. This is essential for mathematical convenience. The resulting loss of information (as to which node it is incident at) is electrically unimportant. The isolated node h corresponds to a zero row. Since the graph is disconnected the columns and rows can be ordered so that the block diagonal nature of the incidence matrix is evident.

Exercise 2.25 *(k) Prove:*
A matrix **K** *is the incidence matrix of some graph* $\mathcal{G}$ *iff it is a* $0, \pm 1$ *matrix and has either zero columns or columns with one* $+1$ *and one* -1 *and remaining entries* 0.

Exercise 2.26 *(k) Prove:*
The sum of the rows of **A** *is* **0**. *Hence the rank of* **A** *is less than or equal to the number of its rows minus* 1.

Exercise 2.27 *(k) Prove:*
If the graph is disconnected the sum of the rows of **A** *corresponding to any component would add up to* **0**. *Hence, the rank of* **A** *is less than or equal to the number of its rows less the number of components (=* $r(\mathcal{G})$*).*

Exercise 2.28 *(k) Prove:*
If $\mathbf{f} = \lambda^{\mathbf{T}}\mathbf{A}$, then $\mathbf{f}(e_i) = \lambda(a) - \lambda(b)$ where a is the positive end point of e_i and b, its negative end point. Thus if λ represents a potential vector with $\lambda(n)$ denoting the potential at n then $\mathbf{f}$ represents the corresponding potential difference vector.

Exercise 2.29 *Construct incidence matrices of various types of graphs e.g. connected, disconnected, tree, circuit, complete graph K_n (every pair of n vertices*
joined by an edge), path.

Exercise 2.30 *Show that the transpose of the incidence matrix of a circuit graph, in which all edges are directed along the orientation of the circuit, is a matrix of the same kind.*

Exercise 2.31 *(k) Show that an incidence matrix remains an incidence matrix under the following operations:*

i. deletion of a subset of the columns,

ii. replacing some rows by their sum.

2.3.1 The Circuit and Crossing Edge Vectors

A **circuit vector** of a graph $\mathcal{G}$ is a vector $\mathbf{f}$ on $E(\mathcal{G})$ corresponding to a circuit of $\mathcal{G}$ with a specified orientation:

$$\begin{aligned} f(e_i) &= +1(-1) \text{ if } e_i \text{ is in the circuit and agrees} \\ &\quad\ \text{with (opposes) the orientation of the circuit.} \\ &= 0 \text{ if } e_i \text{ is not in the circuit.} \end{aligned}$$

Example: The circuit vector associated with the circuit subgraph in Figure 2.6

$$f = \begin{array}{c} \begin{array}{cccccc} e & e_2 & e_3 & e_4 & e_5 & e_6 \end{array} \\ \left[\begin{array}{ccccccccc} -1 & +1 & -1 & +1 & -1 & +1 & 0 & \dots & 0 \end{array} \right] \end{array} \quad (2.2)$$

Exercise 2.32 *(k) Compare a circuit vector with a row of the incidence matrix. Prove:*
A row of the incidence matrix and a circuit vector will

i. have no nonzero entries common if the corresponding node is not present in the circuit subgraph, or

ii. have exactly two nonzero entries common if the node is present in the circuit subgraph. These entries would be ± 1. One of these entries would have opposite sign in the incidence matrix row and the circuit vector and the other entry would be the same in both.

Exercise 2.33 *Prove*

Theorem 2.3.1 *(k) Every circuit vector of a graph $\mathcal{G}$ is orthogonal to every row of the incidence matrix of $\mathcal{G}$.*

(This follows immediately from the statement of the previous exercise). A **crossing edge vector** of a graph $\mathcal{G}$ is a vector $\mathbf{f}$ on $E(\mathcal{G})$ corresponding to a crossing edge set with a specified orientation (V_1, V_2):

$$\begin{aligned} \mathbf{f}(e_i) &= +1(-1) \text{ if } e_i \text{ is in the crossing edge set and agrees} \\ &\quad\; \text{with (opposes) the orientation } (V_1, V_2). \\ &= 0 \text{ if } e_i \text{ is not in the crossing edge set.} \end{aligned}$$

If the crossing edge set is a cutset then the corresponding vector is a **cutset vector**.

Example: The crossing edge vector associated with the crossing edge set of Figure 2.6 is

$$\begin{array}{cccccccc} & e & e_2 & e_3 & e_4 & & & \\ f = \Big[& +1 & -1 & +1 & +1 & 0 & \cdots & 0 \Big]. \end{array} \tag{2.3}$$

Exercise 2.34 *Prove*

Theorem 2.3.2 *(k) The crossing edge vector corresponding to the crossing edge set of orientation (V_1, V_2) is obtained by summing the rows of the incidence matrix corresponding to the nodes in V_1.*

Hence, a crossing edge vector of $\mathcal{G}$ is a voltage vector and is orthogonal to every circuit vector of $\mathcal{G}$. (This can also be proved directly).

Exercise 2.35 *(k) When is a row of the incidence matrix also a cutset vector? Can a cutset be a circuit? Can a cutset vector be a circuit vector?*

Exercise 2.36 *(k)* **RRE of an Incidence Matrix:**
Give a simple rule for finding the RRE of an incidence matrix.

2.3.2 Voltage and Current Vectors

For a graph $\mathcal{G}$ a **current vector** $\mathbf{i}$ is a vector on $E(\mathcal{G})$ that is orthogonal to the rows of the incidence matrix of $\mathcal{G}$, **equivalently**, that satisfies Kirchhoff's current equations (KCE): $\mathbf{Ax} = \mathbf{0}$ [Kirchhoff1847]. A **voltage vector** $\mathbf{v}$ of $\mathcal{G}$ is a vector on $E(\mathcal{G})$ that is linearly dependent on the rows of the incidence matrix of $\mathcal{G}$ i.e.
$\mathbf{v^T} = \lambda^{\mathbf{T}}\mathbf{A}$ for some vector λ.
The vector λ assigns a value to each node of $\mathcal{G}$ and is called a **potential vector**. We say $\mathbf{v}$ is **derived** from the node potential vector λ.
If $\mathbf{c}$ is a circuit vector corresponding to the circuit C with an orientation then the **Kirchhoff's Voltage Equation** (KVE) [Kirchhoff1847] corresponding to C is

$$\mathbf{c}^T\mathbf{x} = 0$$

We now have the following basic characterization of voltage vectors, which is the more conventional way of viewing voltage vectors:

Theorem 2.3.3 *(k) A vector $\mathbf{v}$ on $E(\mathcal{G})$ is a voltage vector iff it satisfies KVE corresponding to each circuit with an orientation.*

Proof : By Theorem 2.3.1 we know that a circuit vector is orthogonal to every row of the incidence matrix. Hence, a circuit vector is orthogonal to any vector that is linearly dependent on the rows of the incidence matrix i.e. orthogonal to a voltage vector. Hence, every voltage vector satisfies KVE corresponding to any circuit with orientation. Now let $\mathbf{v}$ be a vector that satisfies KVE corresponding to every circuit with an orientation. We will construct a potential vector λ s.t. $\lambda^{\mathbf{T}}\mathbf{A} = \mathbf{v^T}$. Take any node d as the datum node, i.e., $\lambda(d) \equiv 0$. Suppose $\lambda(a)$ is already defined and edge e has a as the positive (negative) end and b as the opposite end. Then we take $\lambda(b) \equiv \lambda(a) - v(e) (\lambda(b) \equiv \lambda(a) + v(e))$. In this manner every node in the same connected component is assigned a λ value. A node that is reachable from d by two different paths will not be assigned two different values as otherwise we can find a circuit with orientation for which KVE is violated. Repeating this procedure for each component yields a λ vector s.t. $\lambda^T\mathbf{A} = \mathbf{v^T}$.

□

Voltage vectors and current vectors form vector spaces denoted by

$\mathcal{V}_v(\mathcal{G}), \mathcal{V}_i(\mathcal{G})$, and called voltage space of $\mathcal{G}$ and current space of $\mathcal{G}$ respectively.

Observe that a vector that is orthogonal to the rows of the incidence matrix is also orthogonal to all linear combinations of the rows. Thus $(\mathcal{V}_v(\mathcal{G}))^{\perp}$ is the same as the solution space of Kirchhoff's current equations. We thus have the following celebrated theorem.

Theorem 2.3.4 *(***Tellegen's Theorem (strong form)***)* $(\mathcal{V}_v(\mathcal{G}))^{\perp} = \mathcal{V}_i(\mathcal{G})$.

In the literature, the above theorem is often stated in its 'weak form', viz., that any vector satisfying Kirchhoff's voltage law for a given directed graph is orthogonal to any vector satisfying Kirchhoff's current law for the same graph. In fact, the above theorem says in addition that if a vector is orthogonal to every voltage vector of a graph, it must be a current vector of the same graph.

Remark: When the graph is disconnected with components $\mathcal{G}_1 \dots \mathcal{G}_p$ it is clear that both the current and voltage space can be written in the form $\oplus_{i=1}^{p}\mathcal{V}(\mathcal{G}_i)$. However, in order to write the space in this decomposed form it is not necessary that the $\mathcal{G}_i$ be disconnected. All that is required is that there be no circuit containing edges from different $\mathcal{G}_i$ (see the discussion on separators). We say that graphs $\mathcal{G}_1, \mathcal{G}_2$ are **2-isomorphic** and denote it by $\mathcal{G}_1 \cong \mathcal{G}_2$ iff there exists a bijection $\in: E(\mathcal{G}_1) \rightarrow E(\mathcal{G}_2)$ through which an edge in $\mathcal{G}_1$ can be identified with an edge in $\mathcal{G}_2$ so that $\mathcal{V}_v(\mathcal{G}_1) = \mathcal{V}_v(\mathcal{G}_2)$.
Whitney [Whitney33c] has shown that two 2-isomorphic graphs can be made isomorphic through the repeated use of the following operations:

i. Decompose the graphs into their 2-connected components.

ii. Divide one of the graphs into two subgraphs $\mathcal{G}'$ and $\mathcal{G}$" which have precisely two vertices, say a and b, in common. Split the nodes into a_1, a_2 and b_1, b_2 so that the two subgraphs are now disconnected with a_1, b_1, belonging to $\mathcal{G}'$ and a_2, b_2 to $\mathcal{G}$". Let $\mathcal{G}'_e$ be the graph obtained from $\mathcal{G}'$ by adding an edge e between a_1, b_1. Now reverse all arrows of edges of $\mathcal{G}'$ which lie in the 2-connected component containing e in $\mathcal{G}'_e$ and attach a_1 to b_2 and a_2 to b_1.

2.3.3 Dimension of Voltage and Current Vector Spaces

In this subsection, we compute the rank of $\mathcal{V}_v(\mathcal{G})$ and $\mathcal{V}_i(\mathcal{G})$.

Theorem 2.3.5 *(k) Let G be a graph on n nodes with p connected components. Then*

i. Any set of $(n-p)$ rows of $\mathbf{A}$ which omits one row per component of $\mathcal{G}$, is a basis of $\mathcal{V}_v(\mathcal{G})$.

ii. $r(\mathcal{V}_v(\mathcal{G})) = n - p$

Proof :
If $\mathcal{G}$ is made up of p connected components, by (if necessary) rearranging the rows and columns of $\mathbf{A}$ it can be put in the block diagonal form with p blocks. Hence, any union of linearly independent vectors from different $\mathbf{A}_i$ would be linearly independent. We need to show that dropping any row of $\mathbf{A}_i$ results in a set of linearly independent vectors. So let us, without loss of generality, assume that $\mathcal{G}$ is connected and select any $(n-1)$ rows of $\mathbf{A}$. Suppose these are linearly dependent. Then there is a non trivial linear combination of these rows which is a zero vector. From this set of rows we omit all the rows which are being multiplied by zeros. The remaining set of rows is nonvoid. Consider the corresponding set of vertices say V_1. This set does not contain all vertices of the graph. Since the graph is connected there must be an edge e with one end point in V_1 and the other outside. The submatrix of $\mathbf{A}$ with rows V_1 has only one nonzero entry in the column e. Hence, by multiplying these rows by nonzero scalars and adding we cannot get a zero row. This contradiction shows that any $(n-1)$ rows of $\mathbf{A}$ must be linearly independent. Since the sum of rows of $\mathbf{A}$ is a zero vector, dropping one row of $\mathbf{A}$ results in a basis of $\mathcal{V}_v(\mathcal{G})$ when $\mathcal{G}$ is connected and hence any set of $(n-p)$ rows of $\mathbf{A}$ which omits one row per component of $\mathcal{G}$ is a basis of $\mathcal{V}_v(\mathcal{G})$. Hence, $r(\mathcal{V}_v(\mathcal{G})) = n - p$.

□

A **reduced incidence matrix** $\mathbf{A}_r$ of a graph $\mathcal{G}$ is obtained by omitting one row belonging to each component of $\mathcal{G}$.

We know by Theorem 2.3.5 that the reduced incidence matrix is a representative matrix for $\mathcal{V}_v(\mathcal{G})$. A standard representative matrix for $\mathcal{V}_v(\mathcal{G})$ may be built as described below.

2.3.4 Fundamental cutset matrix of a forest f

We know by Theorem 2.2.7 that there is a unique cutset of a graph $\mathcal{G}$ that intersects a forest f in an edge e. This we have called the fundamental cutset of e with respect to f and denoted it by $B(e, f)$. We assign this cutset an orientation agreeing with that of e. Let $e_1, e_2, \ldots, e_r$ be the edges in the forest f and let $\mathbf{v}_1, \ldots, \mathbf{v}_r$ be the corresponding cutset vectors. A matrix which has $\mathbf{v}_1, \ldots, \mathbf{v}_r$ as rows is called the **fundamental cutset matrix $\mathbf{Q}_f$** of f. This matrix is unique within permutation of rows and columns. By reordering rows and columns, if required, this matrix can be cast in the form

$$\mathbf{Q}_f \equiv \begin{array}{c} \begin{array}{cc} \bar{f} & f \end{array} \\ \left[\begin{array}{cc} \mathbf{Q}_{11} & \mathbf{I} \end{array} \right] \end{array} \tag{2.4}$$

It is clear that $\mathbf{Q}_f$ has $\mid f \mid (= (n-p))$ rows which are linearly independent. Since a cutset vector is linearly dependent on the rows of the incidence matrix $\mathbf{A}$ (Theorem 2.3.2) and $r(\mathbf{A}) = n - p$, it follows that $\mathbf{Q}_f$ is a standard representative matrix for $\mathcal{V}_v(\mathcal{G})$.
Example: Consider the graph of Figure 2.10.
Let $f \equiv \{e_3 \ e_4 \ e_5 \ e_6 \ e_7\}$ and let $\bar{f} = \{e_1 \ \ e_2\}$.

$$\mathbf{Q}_f = \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \left[\begin{array}{ccccccc} 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ -1 & -1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \end{array} \tag{2.5}$$

2.3.5 Fundamental circuit matrix of a forest f

We have already seen that addition of an edge e to a forest f creates a unique circuit which we have called the fundamental circuit of e with

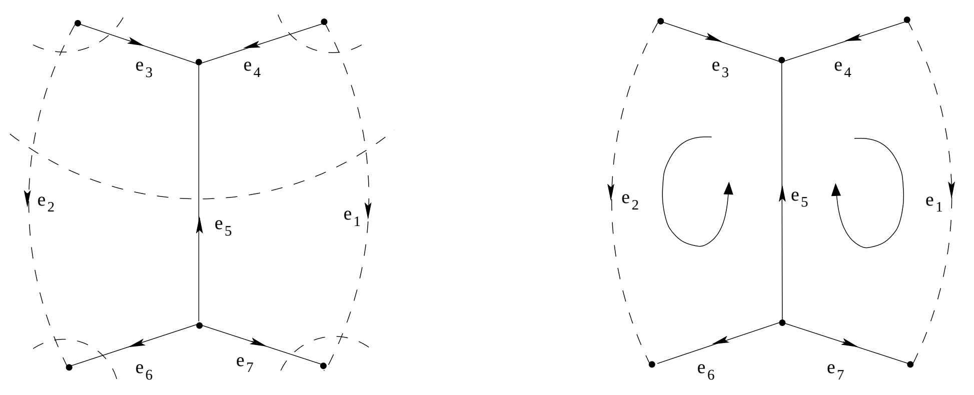


Figure 2.10: f-cutsets and f-circuits

respect to f denoted by $L(e, f)$. As before we assign this circuit an orientation agreeing with that of e. Let $e_1, \cdots, e_\nu$ be edges in the coforest $\bar{f}$. Let $\mathbf{c}_1, \cdots, \mathbf{c}_\nu$ be the corresponding circuit vectors. A matrix with these vectors as rows is called the **fundamental circuit matrix $\mathbf{B}_f$** of f. This matrix is unique within permutation of rows and columns. By reordering rows and columns, if required, this matrix can be cast in the form

$$\mathbf{B}_f \equiv \begin{array}{cc} \bar{f} & f \\ [\mathbf{I} & \mathbf{B}_{12}] \end{array}$$

It is clear that $\mathbf{B}_f$ has $\mid \bar{f} \mid$ rows which are linearly independent. Since a circuit vector is orthogonal to all the rows of the incidence matrix, it must be a current vector. Thus rows of $\mathbf{B}_f$ are current vectors.
Example: Consider the graph in Figure 2.10. Here $f \equiv \{e_3,\ e_4,\ e_5,\ e_6,\ e_7\}$ and $\bar{f} \equiv \{e_1,\ e_2\}$.

$$\mathbf{B}_f = \begin{array}{c} \begin{array}{ccccccc} e_1 & e_2 & e_3 & e_4 & e_5 & e_6 & e_7 \end{array} \\ \left[\begin{array}{ccccccc} 1 & 0 & 0 & -1 & +1 & 0 & -1 \\ 0 & 1 & -1 & 0 & +1 & -1 & 0 \end{array} \right] \end{array}. \qquad (2.6)$$

Theorem 2.3.6 *(k) Let $\mathcal{G}$ be a graph on e edges, n nodes and p connected components. Then $r(\mathcal{V}_i(\mathcal{G})) = e - n + p$.*

Proof : By Theorem 1.2.5, $r(\mathcal{V}_v(\mathcal{G})) + r(\mathcal{V}_v(\mathcal{G}))^\perp = e$
We have already seen that $r(\mathcal{V}_v(\mathcal{G})) = n - p$. Hence $r(\mathcal{V}_v(\mathcal{G}))^\perp =$

$e-n+p$. By Theorem 2.3.4, $(\mathcal{V}_v(\mathcal{G}))^{\perp} = \mathcal{V}_i(\mathcal{G})$. So $r(\mathcal{V}_i(\mathcal{G})) = e-n+p$. We have already seen that $r(\mathcal{V}_v(\mathcal{G})) = n-p$.

□

Corollary 2.3.1 *(k) The rows of an f-circuit matrix of a graph $\mathcal{G}$ form a basis for the current space of $\mathcal{G}$.*

Exercise 2.37 *(k) Examine which potential vectors correspond to a zero voltage vector.*

Exercise 2.38 *Consider the column space $\mathcal{C}(\mathbf{A})$ of $\mathbf{A}$. Show that $(\mathcal{C}(\mathbf{A}))^{\perp}$ is one dimensional if the graph is connected. Hence show that $r(\mathbf{A}) = n-1$.*

Exercise 2.39 *(k) The following is another proof for '$r(\mathbf{A}) = n-1$ if the graph is connected'. If the graph is connected $r(\mathbf{A}) \leq n-1$ since the sum of the rows is zero. But $\mathbf{Q}_f$ has $n-1$ independent rows which are linear combinations of rows of $\mathbf{A}$. Hence $r(\mathbf{A}) = n-1$.*

Exercise 2.40 *An* **elementary vector** *of a vector space is a nonzero vector with minimal support (subset on which it takes nonzero values). Prove*

Theorem 2.3.7 *(k) The circuit vector (cutset vector) is an elementary current vector (elementary voltage vector) and every elementary current vector (elementary voltage vector) is a scalar multiple of a circuit vector (cutset vector).*

Exercise 2.41 *Prove*

Theorem 2.3.8 *(k) A set of columns of $\mathbf{A}$ is linearly independent iff the corresponding edges of the graph do not contain a circuit. A set of columns of $\mathbf{B}_f$ is linearly independent iff the corresponding edges of the graph do not contain a cutset.*

Exercise 2.42 *(k) Every standard representative matrix of $\mathcal{V}_v(\mathcal{G})$ (standard representative matrix of $\mathcal{V}_i(\mathcal{G})$) is a fundamental cutset (fundamental circuit) matrix of $\mathcal{G}$.*

Exercise 2.43 An alternative proof of the strong form of Tellegen's Theorem:
(k) Let $\mathbf{B}_f$, $\mathbf{Q}_f$ be the f-circuit and f-cutset matrix with respect to the same forest. Prove:

i. $\mathbf{B}_f^T\mathbf{Q}_f = \mathbf{0}$

ii. If $\mathbf{B}_f = [\mathbf{I}\ \mathbf{B_{12}}]$ *then* $\mathbf{Q}_f = [-\mathbf{B_{12}^T}\ \mathbf{I}]$. *(Note that this implies Theorem 2.2.8).*

iii. Rows of $\mathbf{B}_f$, $\mathbf{Q}_f$ *are current vectors (voltage vectors). Their ranks add upto* $e(=\mid E(\mathcal{G})\mid)$. *Hence,* $(\mathcal{V}_i(\mathcal{G}))^{\perp} = \mathcal{V}_v(\mathcal{G})$.

Exercise 2.44 *(k) Prove*

Theorem 2.3.9 *(k) The maximum number of independent KVE for a graph is* $r(\mathcal{V}_i(\mathcal{G}))(= e - n + p)$.

2.4 Basic Operations on Graphs and Vector Spaces

In this section, we discuss basic operations on graphs (directed and undirected) which correspond to open circuiting some edges and short circuiting some others. These operations are related to two vector space operations: restriction and contraction. Since real vector spaces are associated primarily with directed graphs, henceforth we deal only with such graphs, but, omit the adjective 'directed'.

2.4.1 Restriction and Contraction of Graphs

Let $\mathcal{G}$ be a graph on the set of edges E and let $T \subseteq E$.

Definition 2.4.1 *The graph* $\mathcal{G}\mathbf{open}(\mathbf{E}-\mathbf{T})$ *is the subgraph of* $\mathcal{G}$ *with* T *as the edge set and* $V(\mathcal{G})$ *as the vertex set. Thus* $\mathcal{G}open(E-T)$ *is obtained by removing (deleting) edges in* $E-T$ *leaving their end points in place.*
The **restriction** *of* $\mathcal{G}$ *to* T*, denoted by* $\mathcal{G}\cdot\mathbf{T}$*, is the subgraph of* $\mathcal{G}$ *obtained by deleting isolated vertices from* $\mathcal{G}open(E-T)$*. Thus,* $\mathcal{G}\cdot T$ *is the subgraph of* $\mathcal{G}$ *on* T*.*
If $\mathcal{G}$ *is directed,* $\mathcal{G}open(E-T), \mathcal{G}\cdot T$*, would be directed with edges retaining the original directions they had in* $\mathcal{G}$*.*

Definition 2.4.2 *The graph* $\mathcal{G}\mathbf{short}\ (\mathbf{E}-\mathbf{T})$ *is built by first building* $\mathcal{G}openT$*. Let* $V_1, \cdots, V_k$ *be the vertex sets of the connected components*

of $\mathcal{G}openT$. The set $\{V_1, \cdots, V_k\}$ is the vertex set and T is the edge set of $\mathcal{G}short\ (E-T)$. An edge $e \in T$ would have V_i, V_j as its end points in $\mathcal{G}short\ (E-T)$ iff the end points of e in $\mathcal{G}$ lie in V_i, V_j. If $\mathcal{G}$ is directed, V_i, V_j would be the positive and negative endpoints of e in $\mathcal{G}short\ (E-T)$ provided the positive and negative end points of e in $\mathcal{G}$ lie in V_i, V_j respectively.
(Thus, $\mathcal{G}short\ (E-T)$ is obtained from $\mathcal{G}$ by short circuiting the edges in $(E-T)$ (fusing their end points) and removing them).
The **contraction** *of $\mathcal{G}$ to T, denoted by $\mathcal{G} \times T$, is obtained from $\mathcal{G}short\ (E-T)$ by deleting the isolated vertices of the latter.*

Example: Consider the graph $\mathcal{G}$ of Figure 2.11.
Let $T = \{e_1, e_6, e_{11}\}$. The graph $\mathcal{G}openT$ is shown in the figure. Graph $\mathcal{G} \cdot (E-T)$ is obtained by omitting isolated vertex v_1 from $\mathcal{G}openT$. Graph $\mathcal{G}short\ (E-T)$ is also shown in the same figure. Graph $\mathcal{G} \times T$ is obtained by omitting the isolated vertex $\{\ v_8, v_9\ \}$ from $\mathcal{G}short\ (E-T)$.

We denote $(\mathcal{G} \times T_1) \cdot T_2$, $T_2 \subseteq T_1 \subseteq E(\mathcal{G})$ by $\mathcal{G} \times T_1 \cdot T_2$ and $(\mathcal{G} \cdot T_1) \times T_2$. $T_2 \subseteq T_1 \subseteq E(\mathcal{G})$ by $\mathcal{G} \cdot T_1 \times T_2$. Graphs denoted by such expressions are called **minors** of $\mathcal{G}$. It can be seen that when a set $A \subseteq E(\mathcal{G})$ is being shorted and a disjoint set $B \subseteq E(\mathcal{G})$, is being opened then the final graph does not depend on the order in which these operations are carried out but only on the sets A and B. Now $\mathcal{G} \times T(\mathcal{G} \cdot T)$ differs from $\mathcal{G}short\ (E-T)$ $(\mathcal{G}open(E-T))$ only in that the isolated vertices are omitted. We thus have the following theorem where equality refers to isomorphism.

Theorem 2.4.1 *(k) Let $\mathcal{G}$ be a graph with $T_2 \subseteq T_1 \subseteq E(G)$. Then*

i. $\mathcal{G} \times T_1 \times T_2 = \mathcal{G} \times T_2$,

ii. $\mathcal{G} \cdot T_1 \cdot T_2 = \mathcal{G} \cdot T_2$,

iii. $\mathcal{G} \times T_1 \cdot T_2 = \mathcal{G} \cdot (E - (T_1 - T_2)) \times T_2$.

Proof : The theorem is immediate when we note that both graphs are obtained by shorting and opening the same sets. In (i) $E - T_2$ is shorted while in (ii) $E - T_2$ is opened. In (iii) $E - T_1$ is shorted and $T_1 - T_2$ is opened.

□

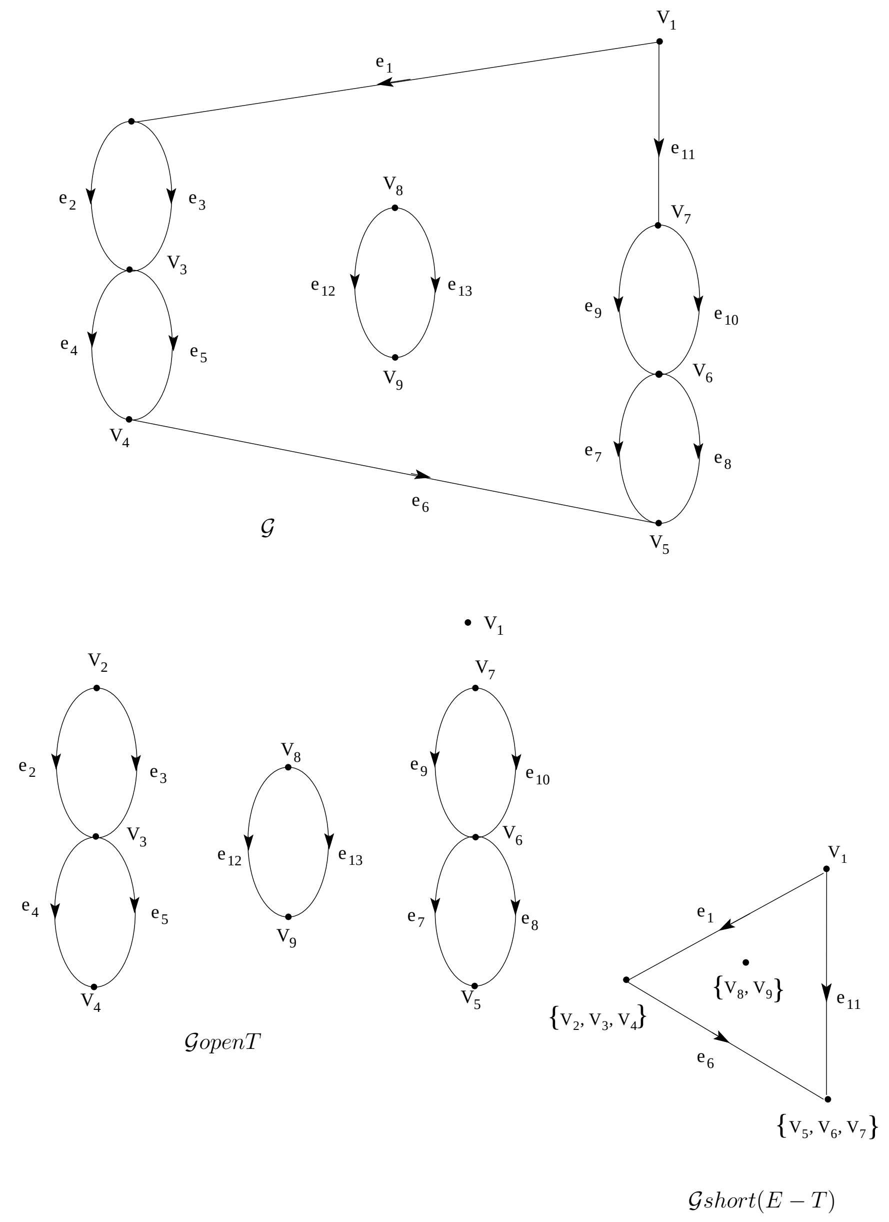


Figure 2.11: Minors of a Graph

Exercise 2.45 *(k)* **Simplification of Expression for minors:**
Show that any minor of the form $\mathcal{G} \times T_1 \cdot T_2 \times T_3 \ldots T_n, T_1 \supseteq T_2 \supseteq \ldots \supseteq T_n$
(the graph being obtained by starting from $\mathcal{G}$ and performing the operations from left to right in succession), can be simplified to a minor of the form
$\mathcal{G} \cdot T^{'} \times T_n$ or $\mathcal{G} \times T^{'} \cdot T_n$.

Exercise 2.46 *Train yourself to visualize $\mathcal{G}_1 \equiv \mathcal{G} short\ (E - T)$ (Put components of $\mathcal{G} open T$ inside surfaces which then become nodes of $\mathcal{G}_1$). How many components does it have? When would a branch of $\mathcal{G}$ become a selfloop of $\mathcal{G}_1$? When would a circuit free set of branches of $\mathcal{G}$ become dependent in $\mathcal{G}_1$?*

Exercise 2.47 Circuits of minors: *Prove*

Lemma 2.4.1 *(k)*

i. A subset C of T is a circuit of $\mathcal{G} \cdot T$ iff C is a circuit of $\mathcal{G}$.

ii. A subset C of T is circuit of $\mathcal{G} \times T$ iff C is a minimal intersection of circuits of $\mathcal{G}$ with T (equivalently,iff C is an intersection of a circuit of $\mathcal{G}$ with T but no proper subset of C is such an intersection).

Exercise 2.48 *(k)* **Cutsets of minors:** *Prove*

Lemma 2.4.2 *(k)*

i. A subset B of T is a cutset of $\mathcal{G} \cdot T$ iff it is a minimal intersection of cutsets of $\mathcal{G}$ with T.

ii. A subset B of T is a cutset of $\mathcal{G} \times T$ iff it is a cutset of $\mathcal{G}$.

2.4.2 Restriction and Contraction of Vector Spaces

We now describe operations on vector spaces which are analogous to the operations of opening and shorting edges in a graph.
Let $\mathcal{V}$ be a vector space on S and let $T \subseteq S$.

Definition 2.4.3 *The* **restriction** *of $\mathcal{V}$ to T, denoted by $\mathcal{V}.T$, is the collection of vectors $\mathbf{f}_T$ where $\mathbf{f}_T$ is the restriction of some vector $\mathbf{f}$ of $\mathcal{V}$ to T.*
The **contraction** *of $\mathcal{V}$ to T, denoted by $\mathcal{V} \times T$, is the collection of vectors $\mathbf{f'}_T$ where $\mathbf{f'}_T$ is the restriction to T of some vector $\mathbf{f}$ of $\mathcal{V}$ such that $\mathbf{f}/(S-T) = \mathbf{0}$.*

It is easily seen that $\mathcal{V} \cdot T$, $\mathcal{V} \times T$ are vector spaces.

As in the case of graphs we denote $(\mathcal{V} \times T_1) \cdot T_2$ by $\mathcal{V} \times T_1 \cdot T_2$. Such expressions denote vector spaces which are called **minors** of $\mathcal{V}$. To bring out the analogy between graph minor and vector space minor operations we say we 'open' T when we restrict $\mathcal{V}$ to $(S-T)$ and say we 'short' T when we contract $\mathcal{V}$ to $(S-T)$.
It turns out that the order in which we open and short disjoint sets of elements is unimportant. More formally we have

Theorem 2.4.2 *(k) Let $T_2 \subseteq T_1 \subseteq S$. Then*

i. $\mathcal{V} \cdot T_1 \cdot T_2 = \mathcal{V} \cdot T_2$,

ii. $\mathcal{V} \times T_1 \times T_2 = \mathcal{V} \times T_2$,

iii. $\mathcal{V} \times T_1 \cdot T_2 = \mathcal{V} \cdot (S - (T_1 - T_2)) \times T_2$.

Proof of (iii): We show that a vector in the LHS belongs to a vector in the RHS.
Let $\mathbf{f}_{T_2} \in \mathcal{V} \times T_1 \cdot T_2$.
Then there exists a vector $\mathbf{f}_{T_1} \in \mathcal{V} \times T_1$ such that $\mathbf{f}_{T_1}/T_2 = \mathbf{f}_{T_2}$ and a vector $\mathbf{f} \in \mathcal{V}$ with $\mathbf{f}/(S-T_1) = \mathbf{0}$ such that $\mathbf{f}/T_1 = \mathbf{f}_{T_1}$.
Now let $\mathbf{f'}$ denote $\mathbf{f}/(S-(T_1-T_2))$.
Clearly $\mathbf{f'} \in \mathcal{V} \cdot (S-(T_1-T_2))$. Now $\mathbf{f'}/(S-T_1) = \mathbf{0}$.
Hence, $\mathbf{f'}/T_2 \in \mathcal{V} \cdot (S-(T_1-T_2)) \times T_2$.
Thus, $\mathcal{V} \times T_1 \cdot T_2 \subseteq \mathcal{V} \cdot (S-(T_1-T_2)) \times T_2$.
The reverse containment is similarly proved.

□

Remark: To see the proof of the above theorem quickly, observe that a typical vector of both LHS and RHS is obtained by restricting a vector of $\mathcal{V}$, that takes zero value on $S-T_1$, to T_2.

Exercise 2.49 *(k) Prove:*
Any minor of the form $\mathcal{V} \times T_1 \cdot T_2 \times T_3 \ldots T_n, T_1 \supseteq T_2 \supseteq \ldots \supseteq T_n$, can be simplified to a minor of the form

$$\mathcal{V} \cdot T' \times T_n \text{ or } \mathcal{V} \times T' \cdot T_n.$$

2.4.3 Vector Space Duality

We now relate the minors of $\mathcal{V}$ to the minors of $\mathcal{V}^{\perp}$. We remind the reader that $\hat{\mathcal{V}}^{\perp}$, the complementary orthogonal space of $\hat{\mathcal{V}}$ is defined to be on the same set as $\hat{\mathcal{V}}$. In the following results we see that the contraction (restriction) of a vector space corresponds to the restriction (contraction) of the orthogonal complement. We say that contraction and restriction are **(orthogonal) duals** of each other.

Theorem 2.4.3 *(k) Let $\mathcal{V}$ be a vector space on S and let $T \subseteq S$. Then,*

i. $(\mathcal{V} \cdot T)^{\perp} = \mathcal{V}^{\perp} \times T$.

ii. $(\mathcal{V} \times T)^{\perp} = \mathcal{V}^{\perp} \cdot T$.

Proof :
i. Let $\mathbf{g}_T \in (\mathcal{V} \cdot T)^{\perp}$. For any $\mathbf{f}$ on S let $\mathbf{f}_T$ denote $\mathbf{f}/T$. Now if $\mathbf{f} \in \mathcal{V}$, then $\mathbf{f}_T \in \mathcal{V} \cdot T$ and $< \mathbf{g}_T, \mathbf{f}_T > = 0$.
Let $\mathbf{g}$ on S be defined by $\mathbf{g}/T \equiv \mathbf{g}_T$, $\mathbf{g}/S - T \equiv \mathbf{0}$. If $\mathbf{f} \in \mathcal{V}$ we have

$$\begin{aligned} < \mathbf{f}, \mathbf{g} > &= < \mathbf{f}_T, \mathbf{g}_T > + < \mathbf{f}_{S-T}, \mathbf{g}_{S-T} > \\ &= 0 + < \mathbf{f}_{S-T}, \mathbf{0}_{S-T} > \\ &= 0. \end{aligned}$$

Thus $\mathbf{g} \in \mathcal{V}^{\perp}$ and therefore, $\mathbf{g}_T \in \mathcal{V}^{\perp} \times T$.Hence, $(\mathcal{V} \,.\, T)^{\perp} \subseteq \mathcal{V}^{\perp} \times T$.
Next let $\mathbf{g}_T \in \mathcal{V}^{\perp} \times T$.
Then there exists $\mathbf{g} \in \mathcal{V}^{\perp}$ s.t. $\mathbf{g}/S - T = \mathbf{0}$ and $\mathbf{g}/T = \mathbf{g}_T$.
Let $\mathbf{f}_T \in \mathcal{V} \cdot T$. There exists $\mathbf{f} \in \mathcal{V}$ s.t. $\mathbf{f}/T = \mathbf{f}_T$.
Now $0 =< \mathbf{f}, \mathbf{g} >=< \mathbf{f}_T, \mathbf{g}_T > + < \mathbf{f}_{S-T}, \mathbf{0}_{S-T} > =< \mathbf{f}_T, \mathbf{g}_T >$.
Hence, $\mathbf{g}_T \in (\mathcal{V} \cdot T)^{\perp}$.
We conclude that
$\mathcal{V}^{\perp} \times T \subseteq (\mathcal{V} \cdot T)^{\perp}$. This proves that $(\mathcal{V} \cdot T)^{\perp} = \mathcal{V}^{\perp} \times T$.

ii. We have $(\mathcal{V}^{\perp} \cdot T)^{\perp} = (\mathcal{V}^{\perp})^{\perp} \times T$.
By Theorem 1.2.5
$((\mathcal{V}^{\perp} \cdot T)^{\perp})^{\perp} = \mathcal{V}^{\perp} \cdot T$ and $(\mathcal{V}^{\perp})^{\perp} = \mathcal{V}$. Hence, $\mathcal{V}^{\perp} \cdot T = (\mathcal{V} \times T)^{\perp}$.

□

The following corollary is immediate.

Corollary 2.4.1 *(k)* $(\mathcal{V} \times P \cdot T)^{\perp} = \mathcal{V}^{\perp} \cdot P \times T, T \subseteq P \subseteq S.$

2.4.4 Relation between Graph Minors and Vector Space Minors

We now show that the analogy between vector space minors and graph minors is more substantial than hitherto indicated - in fact the minors of voltage and current spaces of a graph correspond to appropriate graph minors.

Theorem 2.4.4 *(k) Let $\mathcal{G}$ be a graph with edge set E. Let $T \subseteq E$. Then*

i. $\mathcal{V}_v(\mathcal{G} \cdot T) = (\mathcal{V}_v(\mathcal{G})) \cdot T$

ii. $\mathcal{V}_v(\mathcal{G} \times T) = (\mathcal{V}_v(\mathcal{G})) \times T$

Proof : We remind the reader that by definition a voltage vector $\mathbf{v}$ is a linear combination of the rows of the incidence matrix, the coefficients of the linear combination being given by the entries in a potential vector λ. We say $\mathbf{v}$ is derived from λ.
i. Let $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G} \cdot T)$
Now $\mathcal{V}_v(\mathcal{G} \cdot T) = \mathcal{V}_v(\mathcal{G}open(E - T))$.
Thus, $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G}open(E-T))$. The graph $\mathcal{G}open(E-T)$ has the same vertex set as $\mathcal{G}$ but the edges of $(E - T)$ have been removed.
Let $\mathbf{v}_T$ be derived from the potential vector λ of $\mathcal{G}open(E - T)$. Now for any edge $e \in T$, $\mathbf{v}_T(e) = \lambda(a) - \lambda(b)$, where a, b are the positive and negative end points of e. However, λ is also a potential vector of $\mathcal{G}$. Let the voltage vector $\mathbf{v}$ of $\mathcal{G}$ be derived from λ. For the edge $e \in T$, we have, as before, $\mathbf{v}(e) = \lambda(a) - \lambda(b)$. Thus, $\mathbf{v}_T = \mathbf{v}/T$ and therefore, $\mathbf{v}_T \in (\mathcal{V}_v(\mathcal{G})) \cdot T$. Hence $\mathcal{V}_v(\mathcal{G} \cdot T) \subseteq (\mathcal{V}_v(\mathcal{G})) \cdot T$.
The reverse containment is proved similarly.

ii. Let $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G} \times T)$. Now $\mathcal{V}_v(\mathcal{G} \times T) = \mathcal{V}_v(\mathcal{G}short\ (E-T))$.
Thus, $\mathbf{v}_T \in \mathcal{V}_v(\mathcal{G}short\ (E-T))$.
The vertex set of $\mathcal{G}short\ (E-T)$ is the set $\{V_1, V_2, \ldots V_n\}$ where V_i is the vertex set of the i^{th} component of $\mathcal{G}openT$. Let $\mathbf{v}_T$ be derived from the potential vector $\hat{\lambda}$ in $\mathcal{G}short\ (E-T)$. The vector $\hat{\lambda}$ assigns to each of the V_i the value $\hat{\lambda}(V_i)$. Now define a potential vector λ on the nodes of $\mathcal{G}$ as follows: $\lambda(n) \equiv \hat{\lambda}(V_i), n \in V_i$. Since $\{V_1, \ldots V_k\}$ is a partition of $V(\mathcal{G})$, it is clear that λ is well defined. Let $\mathbf{v}$ be the voltage vector derived from λ in $\mathcal{G}$. Whenever $e \in E-T$ we must have $\mathbf{v}(e) = 0$ since both end points must belong to the same V_i.
Next, whenever $e \in T$ we have $\mathbf{v}(e) = \lambda(a) - \lambda(b)$ where a is the positive end point of e and b, the negative endpoint. Let $a \in V_a$, $b \in V_b$,where $V_a, V_b \in V(\mathcal{G}short\ (E-T))$. Then the positive endpoint of e in $\mathcal{G}short\ (E-T)$ is V_a and the negative end point, V_b.
By definition $\lambda(a) - \lambda(b) = \hat{\lambda}(V_a) - \hat{\lambda}(V_b)$.Thus $\mathbf{v}/T = \mathbf{v}_T$. Hence, $\mathbf{v}_T \in (\mathcal{V}_v(\mathcal{G})) \times T$. Thus, $\mathcal{V}_v(\mathcal{G} \times T) \subseteq (\mathcal{V}_v(\mathcal{G})) \times T$. The reverse containment is proved similarly, but using the idea, that if a voltage vector is zero on all elements of $E-T$, then a potential vector from which it is derived, must have the same value on all vertices of each V_i, since these are vertex sets of componentsof $\mathcal{G}openT$.

□

Using duality we can now prove

Theorem 2.4.5 *(k) Let $\mathcal{G}$ be a directed graph on edge set E. Let $T \subseteq E$. Then,*

i. $\mathcal{V}_i(\mathcal{G}\ .\ T) = (\mathcal{V}_i(\mathcal{G})) \times T$.

ii. $\mathcal{V}_i(\mathcal{G} \times T) = (\mathcal{V}_i(\mathcal{G})) \cdot T$.

Proof :
i. $\mathcal{V}_i(\mathcal{G}\ .\ T) = (\mathcal{V}_v(\mathcal{G}\ .\ T))^\perp$ by the strong form of Tellegen's Theorem.
By Theorem 2.4.4, $\mathcal{V}_v(\mathcal{G}\ .\ T) = (\mathcal{V}_v(\mathcal{G})) \cdot T$.
Hence,

$$\begin{aligned}\mathcal{V}_i(\mathcal{G}\ .\ T) &= ((\mathcal{V}_v(\mathcal{G})) \cdot T)^\perp \\ &= (\mathcal{V}_v(\mathcal{G}))^\perp \times T \\ &= \mathcal{V}_i(\mathcal{G}) \times T.\end{aligned}$$

ii. The proof is similar.

□

Exercise 2.50 *(k) For a connected directed graph $\mathcal{G}$ on node set $\{v_1, \ldots, v_k\}$ if currents $J_1, J_2 \ldots, J_k$ enter nodes $v_1, v_2, \ldots, v_k$ show that there exists a vector $\mathbf{i}$ on $E(\mathcal{G})$, s.t. $\mathbf{Ai} = \mathbf{J}$ iff $\Sigma J_i = 0$.*

Exercise 2.51 *Prove Theorem 2.4.5 directly. (Hint: the result of the preceding exercise would be useful in extending a current vector of $\mathcal{G} \times T$ to a current vector of $\mathcal{G}$).*

2.4.5 Representative Matrices of Minors

As defined earlier, the **representative matrix R** of a vector space $\mathcal{V}$ on S has the vectors of a basis of $\mathcal{V}$ as its rows. Often the choice of a suitable representative matrix would give us special advantages. We describe how to construct a representative matrix which contains representative matrices of $\mathcal{V} \cdot T$ and $\mathcal{V} \times (S - T)$ as its submatrices. We say in such a case that $\mathcal{V} \cdot T$ and $\mathcal{V} \times (S - T)$ become 'visible' in $\mathbf{R}$.

Theorem 2.4.6 *(k) Let $\mathcal{V}$ be a vector space on S. Let $T \subseteq S$. Let $\mathbf{R}$ be a representative matrix as shown below*

$$\begin{array}{cc} & \begin{array}{cc} T & S-T \end{array} \\ \mathbf{R} = & \left[\begin{array}{cc} \mathbf{R}_{TT} & \mathbf{R}_{T2} \\ \mathbf{0} & \mathbf{R}_{22} \end{array} \right] \end{array} \qquad (2.7)$$

where the rows of $\mathbf{R}_{TT}$ are linearly independent. Then $\mathbf{R}_{TT}$ is a representative matrix for $\mathcal{V} \cdot T$ and $\mathbf{R}_{22}$, a representative matrix for $\mathcal{V} \times (S - T)$.

Proof : The rows of $\mathbf{R}_{TT}$ are restrictions of vectors on S to T. Hence, any linear combination of these rows will yield a vector of $\mathcal{V} \cdot T$. If $\mathbf{f}_T$ is any vector in $\mathcal{V} \cdot T$ there exists a vector $\mathbf{f}$ in $\mathcal{V}$ s.t. $\mathbf{f}/T = \mathbf{f}_T$. Now $\mathbf{f}$ is a linear combination of the rows of $\mathbf{R}$. Hence, $\mathbf{f}/T (= \mathbf{f}_T)$ is a linear combination of the rows of $\mathbf{R}_{TT}$. Further it is given that the rows of $\mathbf{R}_{TT}$ are linearly independent. It follows that $\mathbf{R}_{TT}$ is a representative matrix of $\mathcal{V} \cdot T$.

It is clear from the structure of $\mathbf{R}$ (the zero in the second set of rows) that any linear combination of the rows of $\mathbf{R}_{22}$ belongs to $\mathcal{V} \times (S-T)$. Further if $\mathbf{f}$ is any vector in $\mathcal{V}$ s.t. $\mathbf{f}/T = \mathbf{0}$ then $\mathbf{f}$ must be a linear combination only of the second set of rows of $\mathbf{R}$. For, if the first set of rows are involved in the linear combination, since rows of $\mathbf{R}_{TT}$ are linearly independent, $\mathbf{f}/T$ cannot be zero. We conclude that if $\mathbf{f}/(S-T)$ is a vector in $\mathcal{V} \times (S-T)$, it is linearly dependent on the rows of $\mathbf{R}_{22}$. Now rows of $\mathbf{R}$ are linearly independent. We conclude that $\mathbf{R}_{22}$ is a representative matrix of $\mathcal{V} \times T$.

□

Remark: To build a representative matrix of $\mathcal{V}$ with the form as in Theorem 2.4.6, we start from any representative matrix of $\mathcal{V}$ and perform row operations on it so that under the columns T we have a matrix in the RRE form.

The following corollary is immediate

Corollary 2.4.2 *(k)*

$$r(\mathcal{V}) = r(\mathcal{V} \ . \ T) + r(\mathcal{V} \times (S-T)) \ , T \subseteq S$$

Corollary 2.4.3 *(k) Let $\mathcal{G}$ be a graph on E. Then*

$$r(\mathcal{G}) = r(\mathcal{G} \ . \ T) + r(\mathcal{G} \times (E-T)) \ , \forall \ T \subseteq E$$

Proof : We observe that $r(\mathcal{G})$ = number of edges in a forest of $\mathcal{G}$ = $r(\mathcal{V}_v(\mathcal{G}))$. The result follows by Theorem 2.4.4.

□

In the representative matrix of Theorem 2.4.6 the submatrix $\mathbf{R}_{T2}$ contains information about how $T, S-T$ are linked by $\mathcal{V}$. If $\mathbf{R}_{T2}$ is a zero matrix then it is clear that $\mathcal{V} = \mathcal{V}_T \oplus \mathcal{V}_{S-T}$ where $\mathcal{V}_T, \mathcal{V}_{S-T}$ are vector spaces on $T, S-T$.

Definition 2.4.4 *A subset T of S is a* **separator** *of $\mathcal{V}$ iff $\mathcal{V} \times T = \mathcal{V} \ . \ T$.*

It is immediate that if T is a separator so is $(S-T)$. Thus, we might say that $T, (S-T)$ are decoupled in this case. Now by definition $\mathcal{V} \ . \ T \supseteq \mathcal{V} \times T$. Hence, equality of the spaces follows if their dimensions are the same. Hence, T is a separator iff $r(\mathcal{V} \times T) = r(\mathcal{V} \ . \ T)$.

The connectivity of $\mathcal{V}$ at T is denoted by $\xi(T)$ and defined as follows:

$$\xi(T) \equiv r(\mathcal{V} \, . \, T) - r(\mathcal{V} \times T)$$

It is easily seen that $\xi(T) = \xi(S - T)$. Further, this number is zero if T is a separator.

Exercise 2.52 *(k)*

i. Let

$$\mathbf{R} = \begin{array}{c} \begin{array}{ccc} T_1 & T_2 & T_3 \end{array} \\ \left[\begin{array}{ccc} \mathbf{R}_{11} & \mathbf{R}_{12} & \mathbf{R}_{13} \\ \mathbf{R}_{21} & \mathbf{0} & \mathbf{R}_{23} \\ \mathbf{0} & \mathbf{0} & \mathbf{R}_{33} \end{array} \right] \end{array} \qquad (2.8)$$

Rows of $\mathbf{R}_{12}$ *and* $\begin{bmatrix} \mathbf{R}_{11} \\ \mathbf{R}_{21} \end{bmatrix}$ *are given to be linearly independent. Show that* $\mathbf{R}_{33}$ *is a representative matrix of* $\mathcal{V} \times T_3$, $\mathbf{R}_{12}$ *of* $\mathcal{V} \, . \, T_2$, $\mathbf{R}_{21}$ *of* $\mathcal{V} \, . \, (T_1 \cup T_2) \times T_1$ *as well as* $\mathcal{V} \times (T_1 \cup T_3) . \, T_1$ *(and hence these spaces must be the same).*

ii. How would $\mathbf{R}$ *look if* $\mathcal{V} \, . \, (T_1 \cup T_2)$ *has* T_1, T_2 *as separators?*

Exercise 2.53 *(k) Let*

$$\mathbf{R} = \begin{array}{c} \begin{array}{cc} T_1 & T_2 \end{array} \\ \left[\begin{array}{cc} \mathbf{R}_{11} & \mathbf{0} \\ \mathbf{R}_{21} & \mathbf{R}_{22} \\ \mathbf{0} & \mathbf{R}_{33} \end{array} \right] \end{array} . \qquad (2.9)$$

Suppose rows of

$\begin{pmatrix} \mathbf{R}_{11} \\ \mathbf{R}_{21} \end{pmatrix}, \begin{pmatrix} \mathbf{R}_{22} \\ \mathbf{R}_{33} \end{pmatrix}$, *are linearly independent. Show that the number of rows of* $\mathbf{R}_{22} = r(\mathcal{V} \, . \, T_2) - r(\mathcal{V} \times T_2)$ $(= r(\mathcal{V} \, . \, T_1) - r(\mathcal{V} \times T_1))$.

Exercise 2.54 *(k) Prove:*
Let $\xi'(\cdot)$ *be the* $\xi(\cdot)$ *function for* $\mathcal{V}^{\perp}$. *Then* $\xi'(T) = \xi(T), \ \forall \ T \subseteq S$.

Exercise 2.55 *(k) Show that the union of a forest of* $\mathcal{G} \times T$ *and a forest of* $\mathcal{G} \, . \, (E - T)$ *is a forest of* $\mathcal{G}$. *Hence, (Corollary 2.4.3)* $r(\mathcal{G} \times T) + r(\mathcal{G} \, . \, (E - T)) = r(\mathcal{G})$.

Exercise 2.56 *(k) Prove:*
$\nu(\mathcal{G}\ .\ T) + \nu(\mathcal{G} \times (S - T) = \nu(\mathcal{G})$.

Exercise 2.57 *(k) Prove:*
Let $\mathcal{G}$ be a graph on E. Then $T \subseteq E$ is a **separator** *of $\mathcal{G}$ (i.e., no circuit intersects both T and $E - T$ (Subsection 2.2.9) iff T is a separator of $\mathcal{V}_v(\mathcal{G})$. Hence, T is a separator of $\mathcal{G}$ iff $r(\mathcal{G}\ .\ T) = r(\mathcal{G} \times T)$.*

Exercise 2.58 *Let T be a separator of $\mathcal{G}$. Let $\mathcal{G}\ .\ T, \mathcal{G}\ .\ (E-T)$ have α_1, α_2 forests respectively, β_1, β_2 circuits respectively and γ_1, γ_2 cutsets respectively. How many forests, coforests, circuits and cutsets does $\mathcal{G}$ have?*

2.4.6 Minty's Theorem

Tellegen's Theorem is generally regarded as the most fundamental result in Electrical Network Theory. There is however, another fundamental result which can be proved to be formally equivalent to Tellegen's Theorem [Narayanan85c] and whose utility is comparable to the latter. This is Minty's Theorem (strong form) [Minty60], which we state and prove below.

Theorem 2.4.7 *(***Minty's Theorem (strong form)***) Let $\mathcal{G}$ be a directed graph.*
Let $E(\mathcal{G})$ be partitioned into red,blue and green edges. Let e be a green edge.
Then e **either** *belongs to a circuit containing only blue and green edges with all green edges of the same direction with respect to the orientation of the circuit* **or** *e belongs to a cutset containing only red and green edges with all green edges of the same direction with respect to the orientation of the cutset but* **not both**.

Proof: We first prove the weak form:

'in a graph each edge is present in a directed circuit or
in a directed cutset but not both'

Proof of weak form: We claim that a directed circuit and a directed cutset of the same graph cannot intersect. For, suppose otherwise. Let the directed cutset have the orientation (V_1, V_2). The directed circuit subgraph must necessarily have vertices in V_1 as well as in V_2 in order that the intersection be nonvoid. But if we traverse the circuit subgraph starting from the node in V_1 we would at some stage crossover into V_2 by an edge e_{12} and later return to V_1 by an edge e_{21}. Now e_{12}, e_{21} have the same orientation with respect to the circuit which means that if one of them has positive end point in V_1 and negative end point in V_2 the other must have the positive and negative end points in V_2, V_1, respectively. But this contradicts the fact that they both belong to the same directed cutset with orientation (V_1, V_2).

Next we show that any edge e must belong either to a directed circuit or to a directed cutset. To see this, start from the negative end point n_2 of the edge and reach as many nodes of the graph as possible through directed paths. If through one of these paths we reach the positive end point n_1 of e we can complete the directed circuit using e. Suppose n_1 is not reachable through directed paths from n_2. Let the set of all nodes reachable by directed paths from n_2 be enclosed in a surface. This surface cannot contain n_1 and has at least one edge, namely e with one end inside the surface and one outside. It is clear that all such edges must be directed into the surface as otherwise the surface can be enlarged by including more reachable nodes. This collection of edges is a directed crossing edge set and contains a directed cutset which has e as a member (see Exercise 2.59). This completes the proof of the weak form.

Proof of strong form: We open the red edges r and short the blue edges b to obtain from $\mathcal{G}$, the graph $\mathcal{G}_g$ on the green edge set g ,i.e., $\mathcal{G}_g = \mathcal{G} \times (E(\mathcal{G}) - b) \cdot g$. In this graph the weak form holds. Suppose the edge e is part of a directed cutset in $\mathcal{G}_g$. Then this is still a directed cutset containing only green edges in $\mathcal{G} \cdot (E(\mathcal{G}) - r)$. (By Lemma 2.4.2, a set $C \subseteq T \subseteq E(\mathcal{G})$ is a cutset of $\mathcal{G} \times T$ iff it is a cutset of $\mathcal{G}$). It would be a part of a red and green cutset in $\mathcal{G}$ when red edges are introduced between existing nodes. On the other hand, suppose the edge e is part of a directed circuit in $\mathcal{G}_g$. Then this is still a directed circuit containing only green edges in $\mathcal{G} \times (E(\mathcal{G}) - b)$. (By Lemma 2.4.1, a set $C \subseteq T \subseteq E(\mathcal{G})$ is a circuit of $\mathcal{G} \cdot T$ iff it is a circuit of $\mathcal{G}$). It would be a part of a blue and green circuit in $\mathcal{G}$ when blue edges are introduced by splitting existing nodes.
Thus, the strong form is proved.

□

Exercise 2.59 *(k) Let e be a member of a directed crossing edge set C. Show that there exists a directed cutset C_1 s.t. $e \in C_1 \subseteq C$.*

Exercise 2.60 *(k)* **A Generalization:** *Prove:*
Let $\mathcal{V}$ be a vector space on S over the real field and let $e \in S$. Then e is in the support of a nonzero nonnegative vector $\mathbf{f}$ in $\mathcal{V}$ or in the support of a nonzero nonnegative vector $\mathbf{g}$ in $\mathcal{V}^{\perp}$ but not in both.

Exercise 2.61 *(k)* **Partition into strongly connected components:** *Prove:*
The edges of a directed graph can be partitioned into two sets - those that can be included in directed circuits and those which can be included in directed cutsets.

i. Hence show that
the vertex set of a directed graph can be partitioned into blocks so that any pair of vertices in each block are reachable from each other; partial order can be imposed on the blocks s.t. $B_i \geq B_j$ iff a vertex of B_j can be reached from a vertex of B_i.

ii. Give a good algorithm for building the partition as well as the partial order.

2.5 Problems

Problems on Graphs

Problem 2.1 *(k) If a graph has no odd degree vertices,then it is possible to start from any vertex and travel along all edges without repeating any edge and to return to the starting vertex. (Repetition of nodes is allowed).*

Problem 2.2 *(k) Any graph on 6 nodes has either 3 nodes which are pairwise adjacent or 3 nodes which are pairwise non-adjacent.*

Problem 2.3 *(k) A graph is made up of parallel but oppositely directed edges only. Let $T, E - T$ be a partition of the edges of $\mathcal{G}$ such that*

i. *if $e \in T$ then the parallel oppositely directed edge $e' \in T$.*

ii. *it is possible to remove from each parallel pair of edges in $T(E-T)$ one of the edges so that the graph is still strongly connected.*

Show that it is possible to remove one edge from each parallel pair of edges in $\mathcal{G}$ so that the graph remains strongly connected.

Problem 2.4 *(k) We denote by $\mathcal{K}_n$ the graph on n nodes with a single edge between every pair of nodes and by $\mathcal{K}_{m,n}$ the bipartite graph (i.e.,no edges between left vertices and no edges between right vertices) on m left vertices and n right vertices, with edges between every pair of right and left vertices.*

i. *How many edges do $\mathcal{K}_n, \mathcal{K}_{m,n}$ have?*

ii. *Show that every circuit of $\mathcal{K}_{m,n}$ has an even number of edges.*

iii. *Show that $\mathcal{K}_n$ has n^{n-2} trees.*

iv. *A vertex colouring is an assignment of colours to vertices of the graph so that no two of them which have the same colour are adjacent. What is the minimum number of colours required for $\mathcal{K}_n, \mathcal{K}_{m,n}$?*

Problems on Circuits

Problem 2.5 *[Whitney35]* **Circuit Matroid:** *Show that the collection $\mathcal{C}$ of circuits of a graph satisfy the* **matroid circuit axioms***:*

i. *If $C_1, C_2 \in \mathcal{C}$ then C_1 cannot properly contain C_2.*

ii. *If $e_c \in C_1 \cap C_2, e_d \in C_1 - C_2$, then there exists $C_3 \in \mathcal{C}$ and $C_3 \subseteq C_1 \cup C_2$ s.t. $e_c \notin C_3$ but e_d does.*

Problem 2.6 *(k)* **Circuit Characterization:**

i. *A subset of edges C is a circuit of a graph iff it is a minimal set of edges not intersecting any cutset in a single branch.*

ii. *Same as (i) except 'single branch' is replaced by 'odd number of branches'.*

iii. *C is a circuit of a graph iff it is a minimal set of branches not contained in any forest (intersecting every coforest).*

Problem 2.7 *(k)* **Cyclically Connected in terms of Edges:** *A graph in which any two vertices can be included in a circuit subgraph is said to be cyclically connected. In such a graph any two edges can also be so included.*

Problem 2.8 *(k)* **Cut Vertex:** *A graph with no coloops is cyclically connected iff it has no cut vertex (a vertex whose removal along with its incident edges disconnects the graph).*

Problems on Cutsets

Problem 2.9 *(k)* **Cutset Matroid:** *Show that the collection of cutsets of a graph satisfies the circuit axioms of a matroid.*

Problem 2.10 *(k)* **Cutset Characterization:**

i. *A subset of edges C is a cutset of a graph iff it is a minimal set of edges not intersecting any circuit in a single edge (in an odd number of edges).*

ii. *C is a cutset of a graph iff it is a minimal set of branches not contained in any coforest (intersecting every forest).*

Problem 2.11 *(k) Show that every crossing edge set is a disjoint union of cutsets.*

Problem 2.12 *(k)* **Cyclically Connected in terms of Edges in Cutsets:** *In a cyclically connected graph any two edges can be included in a cutset.*

Problems on Graphs and Vector Spaces

Problem 2.13 *(k) Show directly that KCE of a tree graph has only the trivial solution. What is the structure for which KVE has only the trivial solution?*

Problem 2.14 Rank of Incidence Matrix of a Tree Graph:
Give three proofs for 'rank of incidence matrix of a tree graph = number of edges of the graph' using

i. *the determinant of a reduced incidence matrix*

ii. *current injection*

iii. *by assuming branches to be voltage sources and evaluating node potentials.*

Problem 2.15 *(k)* **Nontrivial KCE Solution and Coforest:**
Prove directly that the support of every nonzero solution to KCE meets every coforest. Hence, the rows of an f-circuit matrix of $\mathcal{G}$ span $\mathcal{V}_i(\mathcal{G})$. Hence, $r(\mathcal{V}_i(\mathcal{G})) = e - (v - p)$.

Problem 2.16 *(k)* **Nontivial KVE Solution and Forest:**
Prove directly that the support of every nonzero solution to KVE meets every forest. Hence, the rows of an f-cutset matrix of $\mathcal{G}$ span $\mathcal{V}_v(\mathcal{G})$. Hence, $r(\mathcal{V}_v(\mathcal{G})) = (v - p)$.

Problem 2.17 *(k)* **Determinants of Submatrices of Incidence Matrix:**
The determinant of every submatrix of the incidence matrix $\mathbf{A}$ is 0, ± 1. Hence, this property also holds for every $\mathbf{Q}_f$ and $\mathbf{B}_f$.

Problem 2.18 Interpreting Current Equations:
Let $\mathbf{A}$ be an incidence matrix.

i. *Find one solution to $\mathbf{Ax} = \mathbf{b}$, if it exists, by inspection (giving a current injection interpretation).*

ii. *Find one solution to $\mathbf{A^T y} = \mathbf{v}$ by inspection (using voltage sources as branches).*

Problem 2.19 *i.* *Let $\mathbf{A}$ be the incidence matrix of $\mathcal{G}$. If $\mathbf{Ax} = \mathbf{b}$ is equivalent to $\mathbf{Q}_f\mathbf{x} = \hat{\mathbf{b}}$, relate $\hat{\mathbf{b}}$ to $\mathbf{b}$. Using current injection give a simple rule for obtaining $\hat{\mathbf{b}}$ from $\mathbf{b}$.*

ii. *If $\mathbf{Q}_{f_1}\mathbf{x} = \mathbf{b}_1$, and $\mathbf{Q}_{f_2}\mathbf{x} = \mathbf{b}_2$ are equivalent give a simple rule for obtaining $\mathbf{b}_1$ from $\mathbf{b}_2$.*

iii. If $\mathbf{B}_{f1}\mathbf{y} = \mathbf{d}_1$, and $\mathbf{B}_{f2}\mathbf{y} = \mathbf{d}_2$ are equivalent give a simple rule for obtaining $\mathbf{d}_1$ from $\mathbf{d}_2$.

Problem 2.20 *If two circuit (cutset) vectors figure in the same f-circuit (f-cutset) matrix show that the signs of the overlapping portion fully agree or fully oppose. So overlapping f-circuits (f-cutsets) fully agree or fully oppose in their orientations.*

Problem 2.21 *(k) Give simple rules for computing $\mathbf{A}\mathbf{A}^T, \mathbf{B}_f\mathbf{B}_f^T, \mathbf{Q}_f\mathbf{Q}_f^T$. Show that the number of nonzero entries of $\mathbf{A}\mathbf{A}^T$ is $2e+n$ if the graph has no parallel edges. Show that $\mathbf{B}_f\mathbf{B}_f^T, \mathbf{Q}_f\mathbf{Q}_f^T$ may not have any zero entries. Hence observe that nodal analysis is preferable to fundamental loop analysis and fundamental cutset analysis from the point of view of using Gaussian elimination.*
(Consider the case where a single edge lies in every circuit (cutset) corresponding to rows of $\mathbf{B}_f(\mathbf{Q}_f)$).

Problem 2.22 *Under what conditions can two circuit (cutset) vectors of a given graph be a part of the same f-circuit (f-cutset) matrix?*

Problem 2.23 *(k) Construct good algorithms for building f-circuit and f-cutset vectors for a given forest (use dfs or bfs described in Subsections 2.6.1, 2.6.2). Compute the complexity.*

Problem 2.24 Special Technique for Building a Representative Matrix of $\mathcal{V}_i(\mathcal{G})$:
Prove that the following algorithm works for building a representative matrix of $\mathcal{V}_i(\mathcal{G})$:
Let $\mathcal{G}_1$ be a subgraph of $\mathcal{G}$,
$\mathcal{G}_2$ be a subgraph of $\mathcal{G}$ s.t. $E(\mathcal{G}_1) \cap E(\mathcal{G}_2)$ is a forest of $\mathcal{G}_1$,
$\vdots$
$\mathcal{G}_k$ be a subgraph of $\mathcal{G}$ s.t. $E(\mathcal{G}_k) \cap \left[\bigcup_{i=1}^{k-1} E(\mathcal{G}_i)\right]$ is a forest of the subgraph
$\mathcal{G} \cdot \left(\bigcup_{i=1}^{k-1} E(\mathcal{G}_i)\right)$ and let $\bigcup E(\mathcal{G}_i) = E(\mathcal{G})$.
Build representative matrices $\mathbf{R}_j$ for $\mathcal{V}_i(\mathcal{G}_j), j = 1, 2, \cdots k$. Extend the rows of $\mathbf{R}_j$ to size $E(\mathcal{G})$ by padding with 0s. Call the resulting matrix $\hat{\mathbf{R}}_j$. Then $\mathbf{R}$ is a representative matrix for $\mathcal{V}_i(\mathcal{G})$, where

$$\mathbf{R} = \begin{bmatrix} \hat{\mathbf{R}}_1 \\ \vdots \\ \hat{\mathbf{R}}_k \end{bmatrix}.$$

Problem 2.25 Equivalence of Minty's and Tellegen's Theorems:
Prove that Minty's Theorem (strong form) and Tellegen's Theorem (strong form) are formally equivalent.

Problems on Basic Operations of Graphs

Problem 2.26 *(k) Let $\mathcal{G}$ be graph. Let $K \subseteq E(\mathcal{G})$. Then*

i. *K is a forest of $\mathcal{G} \cdot T$ iff it is a maximal intersection of forests of $\mathcal{G}$ with T.*

ii. *K is a forest of $\mathcal{G} \times T$ iff it is a minimal intersection of forests of $\mathcal{G}$ with T.*

iii. *K is a forest of $\mathcal{G} \times T$ iff $K \cup$ (a forest of $\mathcal{G} \cdot (S - T)$) is a forest of $\mathcal{G}$.*

iv. *K is a coforest of $\mathcal{G} \cdot T$ iff $K \cup$ (a coforest of $\mathcal{G} \times (S - T)$) is a coforest of $\mathcal{G}$.*

Problem 2.27 Relation between Forests Built According to Priority and Graph Minors: *Let $A_1, \cdots A_n$ be pairwise disjoint subsets of $\mathcal{G}$.*

i. *A forest f of $\mathcal{G}$ contains edges from these sets in the same priority iff it is the union of forests from $\mathcal{G} \cdot A_1$, $\mathcal{G} \cdot (A_1 \cup A_2) \times A_2$, $\mathcal{G} \cdot (A_1 \cup A_2 \cup A_3) \times A_3, \cdots \mathcal{G} \times A_n$.*

ii. *Suppose the graph has only such forests what can you conclude?*

iii. *What can you conclude if the priority sequence $A_i, i = 1, \cdots n$ and $A_{\sigma(i)} i = 1, \cdots n$ for every permutation σ of $1, \cdots n$ yield the same forests?*

Problem 2.28 *(k) Show how to build an f-circuit (f-cutset) matrix of $\mathcal{G}$ in which f-circuit (f-cutset) matrices of $\mathcal{G} \cdot T$ and $\mathcal{G} \times (E - T)$ become 'visible' (appear as submatrices). Let $T_2 \subseteq T_1 \subseteq E(\mathcal{G})$. Repeat the above so that the corresponding matrix of $\mathcal{G} \times T_1 \cdot T_2$ is 'visible'.*

Problem 2.29 *(k) Suppose in an electrical network on graph $\mathcal{G}$ the subset T is composed of current (voltage) sources. How will you check that they do not violate KCL (KVL)?*

2.6 Graph Algorithms

In this section we sketch some of the basic graph algorithms which we take for granted in the remaining part of the book. The algorithms we consider are

- construction of trees and forests of various kinds for the graph (bfs, dfs, minimum spanning)
- finding the connected components of the graph
- construction of the shortest path between two vertices of the graph
- construction of restrictions and contractions of the graph
- bipartite graph based algorithms such as for dealing with partitions
- flow maximization in networks

The account in this section is very brief and informal. For more details the readers are referred to [Aho+Hopcroft+Ullman74] [Kozen92] [Cormen+Leiserson+Rivest90].

For each of the above algorithms we compute or mention the 'asymptotic worst case complexity' of the algorithm. Our interest is primarily in computing an upper bound for the worst case running time of the algorithm and sometimes also for the worst case storage space required for the algorithm. A memory unit, for us, contains a single elementary

symbol (a number - integer or floating point, or an alphabet). Accessing or modifying such a location would be assumed to cost unit time. Operations such as comparison, addition, multiplication and division are all assumed to cost unit time. Here as well as in the rest of the book we use the 'big Oh' notation:
Let $f, g : \mathcal{N}^p \rightarrow \mathcal{N}$ where $\mathcal{N}$ denotes the set of nonnegative integers and p is a positive integer. We say f **is** $O(g)$ iff there exists a positive integer k s.t. $f(\mathbf{n}) \leq kg(\mathbf{n})$ for all $\mathbf{n}$ outside a finite subset of $\mathcal{N}^p$.

The time and space complexity of an algorithm to solve the problem (the number of elementary steps it takes and the number of bits of memory it requires) would be computed in terms of the **size of the problem instance**. The size normally refers to the number of bits (within independently specified multiplying constant) required to represent the instance of the problem in a computer. It could be specified in terms of several parameters. For example, in the case of a directed graph with capacitated edges the size would be in terms of number of vertices, number of edges and the maximum number of bits required to represent the capacity of an edge. In general, the **size of a set** would be its cardinality while the **size of a number** would be the number of bits required to represent it. Thus, if n is a positive integer, $\log n$ would be its size – the base being any convenient positive integer. All the algorithms we study in this book are polynomial time (and space) algorithms, i.e., their worst case complexity can be written in the form $O(f(n_1, \cdots, n_p))$ where $f(\cdot)$ is a polynomial in the n_i. Further, in almost all cases, the polynomials would have low degree (≤ 5).

Very rarely we have used words such as NP-complete and NP-Hard. Informally, a problem is in P if the 'answer to it' (i.e., the answer to every one of its instances) can be **computed** in polynomial time (i.e., in time polynomial in the size of the instance) and is in NP if the correctness of the candidate answer to every instance of it can be **verified** in polynomial time. It is clear that P $\subseteq$ NP. However, although it is widely believed that P $\neq$ NP, a proof for this statement has not been obtained so far. An **NP-Hard** problem is one which has the property that if its answer can be **computed** in polynomial time, then we can infer that the answer to every problem in NP can be computed in polynomial time. An NP-Hard problem need not necessarily be in NP.

If it is in NP, then it is said to be **NP-complete**. The reader interested in formal definitions as well as in additional details is referred to [Garey+Johnson79], [Van Leeuwen90].

Exercise 2.62 *A decision problem is one for which the answer is (***yes** *or* **no***). Convert the problem 'find the shortest path between* v_1 *and* v_2 *in a graph' into a 'short' sequence of decision problems.*

For most of our algorithms elementary **data structures** such as arrays, stacks, queues are adequate. Where more sophisticated data structures (such as Fibonacci Heaps) are used, we mention them by name and their specific property (such as time for retrieval, time for insertion etc.) that is needed in the context. Details are skipped and may be found in [Kozen92].

Storing a graph: A graph can be stored in the form of a sequence whose i^{th} (composite) element contains the information about the i^{th} edge (names of end points; if edge is directed the names of positive and negative end points). This sequence can be converted into another whose i^{th} element contains the information about the i^{th} node (names of incident edges, their other end points; if the graph is directed, the names of out-directed and in-directed edges and their other end points.) We will assume that we can retrieve incidence information about the i^{th} edge in $O(1)$ time and about the (i^{th} node) in $O(degree\ of\ node\ i)$ time. The conversion from one kind of representation to the other can clearly be done in $O(m+n)$ time where m is the number of edges and n is the number of vertices.

Sorting and Searching: For sorting a set of indexed elements in order of increasing indices, there are available, algorithms of complexity $O(n \log n)$, where n is the number of elements [Aho+Hopcroft+Ullman74]. We use such algorithms without naming them. In such a sorted list of elements to search for a given indexed element takes $O(\log n)$ steps by using **binary search**.

2.6.1 Breadth First Search

A **breadth first search (**bfs**) tree or forest** for the given graph $\mathcal{G}$ is built as follows:
Start from any vertex v_o and **scan** edges incident on it.

Select these edges and put the vertices $v_1, v_2 \cdots v_{ko}$ which are adjacent to v_o in a **queue** in the order in which the edges between them and v_o were scanned.
Mark v_o as belonging to component 1 and level 0. Mark $v_1, \cdots, v_{ko}$, as belonging to component 1 and level 1 and as **children** of v_o. Mark the vertex v_o additionally as a **parent** of its children (against each of its children).
Suppose at any stage we have the queue $v_{i1}, \cdots, v_{ik}$ and a set M_i of marked vertices.
Start from the left end (first) of the queue, scan the edges incident on it and select those edges whose other ends are unmarked. If a selected edge is between v_{ij} and the unmarked vertex v_{um} then the former (latter) is the **parent (child)** of the latter (former).
Put the children of v_{i1} in the queue after v_{ik} and delete v_{i1} from the queue.
Mark these vertices as belonging to the level next to that of v_{i1} and to the same component as v_{i1} and as children of v_{i1} (against v_{i1}). Mark the vertex v_{i1} as a parent of its children (against its children).
Continue.
When the graph is disconnected it can happen that the queue is empty but all vertices have not yet been marked. In this case continue the algorithm by picking an unmarked vertex.
Mark it as of level 0 but as of component number one more than that of the previous vertex. Continue.
STOP when all vertices of the graph have been marked.

At the conclusion of the above algorithm we have a breadth first search forest made up of the selected edges and a partition of the vertex set of the graph whose blocks are the vertex sets of the components of the graph. The starting vertices in each component are called **roots**. The level number of each vertex gives its **distance** from the root (taking the length of each edge to be one). The path in the forest from a given vertex in a component to the root in the component is obtained by travelling from the vertex to its parent and so on back to the root.

In a directed graph a bfs starting from any vertex would yield all vertices reachable from it through directed paths. In this case, while processing a vertex, one selects only the outward directed edges.

The **complexity of the bfs algorithm** is $O(m + n)$ where m

is the number of edges and n is the number of vertices. (Each edge is 'touched' atmost twice. Each vertex other than the root is touched when an edge incident on it is touched or when it is a new root. Except where the root formation is involved the labour involved in touching a vertex can always be absorbed in that of touching an edge. Each touching involves a fixed number of operations).

The **complexity** of computing **all the reachable vertices** from a given vertex or a set of vertices of a directed graph through bfs is clearly also $O(m+n)$.

2.6.2 Depth First Search

A **depth first search** (dfs) **tree or forest** for the given graph $\mathcal{G}$ is built as follows:
Start from any vertex v_o and **scan** the edges incident on it.
Select the first nonselfloop edge. Let v_1 be its other end point. Put v_o, v_1 in a **stack**. (A **stack** is a sequence of data elements in which the last (i.e., latest) element would be processed first). Mark v_o as belonging to component 1 and as having dfs number 0, v_1 as belonging to component 1 and as having dfs number 1. Mark v_o as the parent of v_1 (against v_1) and v_1 as a child of v_o (against v_o).
Suppose at any stage, we have the stack $v_{i1}, \cdots, v_{ik}$ and a set M_i of marked vertices.
Start from the top of the stack, i.e., from v_{ik} and scan the edges incident on it. Let e be the first edge whose other end point v_{i+1} is unmarked. Select e. Mark v_{i+1} as of dfs number one more than that of the highest dfs number of a vertex in M_i and of component number same as that of v_{ik}. Mark (against v_{i+1}) v_{ik} as its parent and (against v_{ik}) v_{i+1} as one of its children. Add v_{i+1} to the top of the stack and repeat the process.
Suppose v_{ik} has no edges incident whose other end points are unmarked. Then delete v_{ik} from the stack (so that $v_{i(k-1)}$ goes to the top of the stack).
Continue.
STOP when all vertices in the graph have been marked.
When the graph is disconnected it can happen that the stack is empty

but all vertices have not yet been marked. In this case continue the algorithm by picking an unmarked vertex. Give it a dfs number 0 but component number one more than that of the previous vertex.

At the conclusion of the above algorithm we have a depth first search forest made up of the selected edges and a partition of the vertex set of the graph whose blocks are the vertex sets of the components of the graph. The starting vertices in each component are called roots. The path in the forest from a given vertex in a component to the root in the component is obtained by travelling from the vertex to its parent and so on back to the root. The **time complexity of the dfs algorithm** can be seen to be $O(m+n)$ where m is the number of edges and n, the number of vertices in the graph.

Exercise 2.63 *(k) Let e be an edge outside a dfs tree of the graph. Let v_1, v_2 be the end points of e with dfs numbering a, b respectively. If $b > a$ show that v_1 is necessarily an ancestor of v_2 (ancestor $\equiv$ parent's parent's ... parent).*

The dfs tree can be used to detect 2-connected components of the graph in $O(m+n)$ time [Aho+Hopcroft+Ullman74]. It can be used to construct the planar embedding of a planar graph in $O(n)$ time [Hopcroft+Tarjan74], [Kozen92]. There is a directed version of the dfs tree using which a directed graph can be decomposed into strongly connected components (maximal subsets of vertices which are mutually reachable by directed paths). Using the directed dfs tree this can be done in $O(m+n)$ time [Aho+Hopcroft+Ullman74].

Fundamental circuits: Let t be a forest of graph $\mathcal{G}$ and let $e \in (E(\mathcal{G}) - t)$. To construct $L(e,t)$ we may proceed as follows: Do a dfs of $\mathcal{G} \,.\, t$ starting from any of its vertices. This would give a dfs number to every vertex in $\mathcal{G} \,.\, t$.
Let v_1, v_2 be the end points of e. From v_1, v_2 proceed towards the root by moving from child to parent until you meet the first common ancestor v_3 of v_1 and v_2. This can be done as follows: Suppose v_1 has a higher dfs number than v_2. Move from v_1 to root until you reach the first v_1' whose dfs number is less or equal to that of v_2. Now repeat the procedure with v_2, v_1' and so on alternately until the first common vertex is reached. This would be v_3. Then $L(e,t) \equiv \{e\} \cup \{$ edges in paths from v_1 to v_3 and v_2 to $v_3\}$.
To build the circuit vector corresponding to $L(e,t)$ proceed as follows:

Let v_1 be the positive end point and v_2, the negative end point of e. The path from v_2 to v_1 in the tree is the path from v_2 to v_3 followed by the path from v_3 to v_1. The circuit vector has value $+1$ at e, 0 outside $L(e,t)$ and $+1$ (-1) at e_j, if it is along (against) the path from v_2 to v_1 in the tree. Complexity of building the $L(e,t)$ is $O(\mid L(e,t) \mid)$ and that of building all the $L(e_i,t)$ is $O(\sum \mid L(e,t) \mid)$.

Exercise 2.64 *How would you build the f-circuit for a bfs tree?*

2.6.3 Minimum Spanning Tree

We are given a connected undirected graph $\mathcal{G}$ with real weights $(w(\cdot))$ on its edges. The problem is to find a spanning tree of least total weight (total weight = sum of weights of edges in the tree). We give **Prim's algorithm** for this purpose:
Choose an arbitrary vertex v_o. Among the edges incident on v_o select one of least weight.
Suppose at some stage, X is the set of edges selected and $V(X)$, the set of their end points. If $V(X) \neq V(\mathcal{G})$, select an edge e of least weight among those which have only one end point in $V(X)$.
Now replace X by $X \cup e$ and repeat.
Stop when $V(X) = V(\mathcal{G})$.
The selected edges constitute a minimum spanning tree.

Exercise 2.65 *Justify Prim's algorithm for minimum spanning tree.*

Complexity: Let n be the number of vertices and m, the number of edges of the graph. The algorithm has n stages. At each stage we have to find the minimum weight edge among the set of edges with one end point in $V(X)$. Such edges cannot be more than m in number. So finding the minimum is $O(m)$ and the overall complexity is $O(mn)$. However, this complexity can be drastically improved if we store the vertices in $(V(\mathcal{G}) - V(X))$ in a Fibonacci Heap. This data structure permits the extraction of the minimum valued element in $O(\log n)$ amortized time (where n is the number of elements in the heap), changing the value of an element in $O(1)$ amortized time and deleting the minimum element in $O(\log n)$ amortized time. (Loosely, an operation being of amortized time $O(f(n))$ implies that, if the entire running of the algorithm involves performing the operation k times, then the time for performing these operations is $O(kf(n))$.

For each vertex v in $(V(\mathcal{G}) - V(X))$ the **value** is the minimum of the weights of the edges connecting it to $V(X)$. To pick a vertex of least value we have to use $O(\log n)$ amortized time. Suppose v has been added to $V(X)$ and X replaced by $X \cup e$, where e has v as one its ends. Now the value of a vertex v' in $(V(\mathcal{G}) - (V(X) \cup e))$ has to be updated only if there is an edge between v and v'. Throughout the algorithm this updating has to be done only once per edge and each such operation takes $O(1)$ amortized time. So overall the updating takes $O(m)$ time. The extraction of the minimum valued element takes $O(n \log n)$ time over all the n stages. At each stage the minimum element has to be deleted from the heap. This takes $O(\log n)$ amortized time and $O(n \log n)$ time overall. Hence, the running time of the algorithm is $O(m + n \log n)$. (Note that the above analysis shows that, without the use of the Heap, the complexity of Prim's algorithm is $O(n^2)$).

2.6.4 Shortest Paths from a Single Vertex

We are given a graph $\mathcal{G}$, without parallel edges, in which each edge e has a nonnegative **length** $l(v_1, v_2)$, where v_1, v_2 are the end points of e. If $v_1 = v_2$, then $l(v_1, v_2) \equiv 0$. The **length of a path** is defined to be the sum of the lengths of the edges in the path.
The problem is to find shortest paths from a given vertex (called the source) to every vertex in the same connected component of the graph.
We give **Dijkstra's Algorithm** for this problem.
Start from the source vertex v_o and assign to each adjacent vertex v_i, a **current distance** $d_c(v_i) \equiv l(v_o, v_i)$. **Mark**, against each v_i, the vertex v_o as its foster parent. (We will call v_i, the foster children of v_o).
Let v_1 be the adjacent vertex to v_o with the least value of $d_c(v_i)$. Declare the **final distance of** v_1, $d_f(v_1) \equiv d_c(v_1)$. Mark, against v_1, the vertex v_o as its parent. (We will call v_1, a child of v_o).
(At this stage we have **processed** v_o and **marked** its adjacent vertices).
Assign a current distance ∞ to each unmarked vertex.
Suppose $X \subseteq V(\mathcal{G})$ denotes the processed set of vertices at some stage.
For each neighbour $v_j \in (V(\mathcal{G}) - X)$ of last added vertex v_k,

Check if $d_c(v_j) > d_f(v_k) + l(v_k, v_j)$.

```
If Yes, then
        Mark, against $v_j$, the vertex $v_k$ as its foster parent
        (deleting any earlier mark, if present). (We will call $v_j$, a foster child of $v_k$).
        Set $d_c(v_j) \equiv d_f(v_k) + l(v_k, v_j)$.
Find a vertex $v_q \in (V(\mathcal{G}) - X)$ with the least current distance $d_c(v_q)$.
Declare $v_q$ to have been processed and its final distance $d_f(v_q)$ from $v_o$ to be $d_c(v_q)$. Mark, against $v_q$, its foster parent $u_q$ as its parent (we will call $v_q$ a child of $u_q$).
Add $v_q$ to $X$. Repeat the procedure with $X \cup v_q$ in place of $X$.
STOP when all vertices in the connected component of $v_o$ are processed.
```

To find a shortest path from a processed vertex v_j to v_o, we travel back from v_j to its parent and so on, from child to parent, until we reach v_o.

Justification: To justify the above algorithm, we need to show that the shortest distance from v_o to v_q (the vertex with the least current distance in $V(\mathcal{G}) - X$) is indeed $d_f(v_q)$. First, we observe that a finite $d_c(v)$, and therefore $d_f(v)$, for any vertex v is the length of some path from v_o to v. By induction, we may assume that for every vertex v_{in} in X, $d_f(v_{in})$ = length of the shortest path from v_o to v_{in}. Note that this is justified when $X = \{v_o\}$. Suppose $d_f(v_q)$ is greater than the length of a shortest path $P(v_o, v_q)$ from v_o to v_q. Let $P(v_o, v_q)$ leave X for the first time at v_3 and let the next vertex be $v_{out} \in (V(\mathcal{G}) - X)$. If $v_{out} = v_q$, we must have
$d_f(v_q) \le d_f(v_3) + l(v_3, v_q) =$ length of $P(v_o, v_q)$.
This is a contradiction. So $v_{out} \neq v_q$. Now
$d_c(v_{out}) \le (d_f(v_3) + l(v_3, v_{out})) \le$ length of $P(v_o, v_q)$.
Hence, $d_c(v_{out}) < d_c(v_q) = d_f(v_q)$, which contradicts the definition of v_q. We conclude that $d_f(v_q)$ must be the length of the shortest path from v_o to v_q.

Complexity: Let n be the number of vertices and m, the number of edges of the graph. This algorithm has n stages. At each stage we have to compute $d_c(v_j)$ for vertices v_j adjacent to the last added vertex. This computation cannot exceed $O(m)$ over all the stages. Further at each stage we have to find the minimum of $d_c(v_i)$ for each v_i in $(V(\mathcal{G}) - X)$. This is $O(n)$. So we have an overall complexity of $O(n^2 + m)$. Now

$m \leq n^2$. So the time complexity reduces to $O(n^2)$.
We note that the complexity of this algorithm reduces to $O(m+n\log n)$ if the elements in $V(\mathcal{G}) - X$ are stored in a Fibonacci Heap (see [Kozen92]).

2.6.5 Restrictions and Contractions of Graphs

Let $\mathcal{G}$ be a graph and let $T \subseteq E(\mathcal{G})$. To build $\mathcal{G} \cdot T$, we merely pick out the edge - end point list corresponding to T. This has complexity $O(\mid T \mid)$. (Note that the edges of T still bear their original index as in the sequence of edges of $\mathcal{G}$).
To build $\mathcal{G}\times T$ we first build $\mathcal{G}openT$. The graph $\mathcal{G}openT$ has $\mathcal{G} \cdot (E(\mathcal{G})-T)$ + remaining vertices of $\mathcal{G}$ as isolated vertices. Next we find the connected components of $\mathcal{G}openT$. Let the vertex sets of the components be $X_1, \cdots, X_k$. For each X_i, whenever $v \in X_i$, mark it as belonging to X_i (some one vertex of X_i can represent X_i). Changing the names of endpoints amounts to directing a pointer from vertices to the X_i that they belong to. Now in the edge - end point list of T, for each edge e, if v_1, v_2 are its (positive and negative) endpoints, and if $v_1 \in X_i$, $v_2 \in X_j$, then replace v_1 by X_i and v_2 by X_j ($\mathcal{G}shortT$ has vertex set $(X_1, \cdots, X_k)$).
The complexity of building $\mathcal{G}openT$ is $O(n+ \mid E-T \mid)$, where n is the number of vertices of $\mathcal{G}$, that of finding its components is $O(n+ \mid E-T \mid)$ (using dfs say). Changing the names of endpoints amounts to directing a pointer from vertices to the X_i that they belong to. This has already been done. So the overall complexity is $O(n+m)$ where $m = \mid E(\mathcal{G}) \mid$.

Elsewhere, we describe methods of network analysis (by decomposition) which require the construction of the graphs $\mathcal{G} \cdot E_1, \cdots, \mathcal{G} \cdot E_k$ or $\mathcal{G} \times E_1, \cdots, \mathcal{G} \times E_k$, where $\{E_1, \cdots, E_k\}$ is a partition of $E(\mathcal{G})$. The complexity of building $\oplus_i \mathcal{G} \cdot E_i$ is clearly $O(n+m)$, while that of building $\oplus_i \mathcal{G} \times E_i$ is $O(k(n+m))$.

2.6.6 Hypergraphs represented by Bipartite Graphs

Hypergraphs are becoming increasingly important for modeling many engineering situations. By definition, a **hypergraph** $\mathcal{H}$ is a pair

$(V(\mathcal{H}), E(\mathcal{H}))$, where $V(\mathcal{H})$ is the set of **vertices** of $\mathcal{H}$ and $E(\mathcal{H})$, a family of subsets of $V(\mathcal{H})$ called the **hyperedges** of $\mathcal{H}$. (We remind the reader that in a family, the same member subset could be repeated with distinct indices yielding distinct members of the family). The reader would observe that undirected graphs are a special case of hypergraphs (with the hyperedges having cardinality 1 or 2). The most convenient way of representing a hypergraph is through a **bipartite graph** $B \equiv (V_L, V_R, E)$ - a graph which has a **left vertex set** V_L, a (disjoint) **right vertex set** V_R and the set of edges E each having one end in V_L and the other in V_R. We could represent $\mathcal{H}$ by $B_{\mathcal{H}} \equiv (V_L, V_R, E)$ identifying $V(\mathcal{H})$ with $V_L, E(\mathcal{H})$ with V_R with an edge in the bipartite graph between $v \in V_L$ and $e \in V_R$ iff v is a member of the hyperedge e of $\mathcal{H}$.

We can define **connectedness** for $\mathcal{H}$ in a manner similar to the way the notion is defined for graphs. $\mathcal{H}$ is **connected** iff for any pair of vertices v_1, v_f there exists an **alternating sequence** $v_1, e_1, v_2, e_2, \cdots, e_f, v_f$, where the v_i are vertices and e_i, edges s.t. each edge has both the preceding and succeeding vertices as members. It is easily seen that $\mathcal{H}$ is connected iff $B_{\mathcal{H}}$ is connected. Hence, checking connectedness of $\mathcal{H}$ can be done in $O(\mid V_L \mid + \mid V_R \mid + \mid E \mid)$ time. Since everything about a hypergraph is captured by a bipartite graph we confine our attention to bipartite graphs in this book. The reader interested in 'standard' hypergraph theory is referred to [Berge73].

2.6.7 Preorders and Partial Orders

A preorder is an ordered pair $(P, \preceq)$ where P is a set and '$\preceq$' is a binary relation on P that satisfies the following:
$x \preceq x, \forall\, x \in P$;
$x \preceq y, y \preceq z \Rightarrow x \preceq z\ , \forall\, x, y, z \in P$.
We can take the elements of P to be vertices and join x and y by an edge directed from y to x if $x \preceq y$. Let $\mathcal{G}_p$ be the resulting directed graph on the vertex set P. Then the vertex sets of the strongly connected components of $\mathcal{G}_P$ are the equivalence classes of the preorder (x, y belong to an equivalence class iff $x \preceq y$ and $y \preceq x$).
Let $\mathcal{P}$ be the collection of equivalence classes. If $X_1, X_2 \in \mathcal{P}$, we define $X_1 \leq X_2$ iff in the graph $\mathcal{G}_p$, a vertex in X_1 can be reached from a

vertex in X_2. It is easily seen that this defines a **partial order** ($X_i \leq X_i; X_i \leq X_j$ and $X_j \leq X_i$ iff $X_i = X_j$; $X_i \leq X_j, X_j \leq X_k \Rightarrow X_i \leq X_k$). This partial order $(\mathcal{P}, \leq)$ is said to be **induced by** $(P, \preceq)$. By using a directed dfs forest on the graph $\mathcal{G}_P$ representing the preorder $(P, \preceq)$ we can get a graph representation of the induced partial order in time $O(m + n)$ where m is the number of edges and n is the number of vertices in $\mathcal{G}_P$ [Aho+Hopcroft+Ullman74].

A partial order can be represented more economically by using a Hasse Diagram. Here a directed edge goes from a vertex y to a vertex x iff y **covers** x, i.e., $x \leq y, x \neq y$ and there is no z s.t. $z \neq x$ and $z \neq y$ and $x \leq z \leq y$. An **ideal** I of $(\mathcal{P}, \leq)$ is a collection of elements of $\mathcal{P}$ with the property that if $x \in I$ and $y \leq x$ then $y \in I$. The **principal ideal** I_x in $(\mathcal{P}, \leq)$ of an element $x \in \mathcal{P}$ is the collection of all elements $y \in \mathcal{P}$ s.t. $y \leq x$. Clearly an ideal is the union of the principal ideals of its elements. A **dual ideal** I^d is a subset of $\mathcal{P}$ with the property that if $x \in I$ and $x \leq z$ then $z \in I^d$. **Ideals and dual ideals of preorders** are defined similarly. The **dual** of a partial order $(\mathcal{P}, \leq)$ is the partial order $(\mathcal{P}, \geq)$, where $x \geq y$ iff $y \leq x$. We define the dual of a preorder in the same manner. We use $\leq$ and $\geq$ interchangeably (writing $y \leq x$ or $x \geq y$) while speaking of a partial order or a preorder.

Preorders and partial orders are used repeatedly in this book (see for instance Chapter **??**).

Lattices

Let $(\mathcal{P}, \leq)$ be a partial order. An **upper bound** of $e_1, e_2 \in \mathcal{P}$ is an element $e_3 \in \mathcal{P}$ s.t. $e_1 \leq e_3$ and $e_2 \leq e_3$.
A **lower bound** of e_1 and e_2 would be an element $e_4 \in \mathcal{P}$ s.t. $e_4 \leq e_1$ and $e_4 \leq e_2$.
A **least upper bound** (l.u.b.) of e_1, e_2 would be an upper bound e^u s.t. whenever e_3 is an upper bound of e_1, e_2 we have $e_3 \geq e^u$. A **greatest lower bound** (g.l.b.) of e_1, e_2 would be a lower bound e_l s.t. whenever e_4 is a lower bound of e_1, e_2 we have $e_4 \leq e_l$. It is easy to see that if l.u.b. (g.l.b.) of e_1, e_2 exists, then it must be unique. We denote the l.u.b. of e_1, e_2 by $e_1 \vee e_2$ and call it the **join** of e_1 and e_2. The g.l.b. of e_1, e_2 is denoted by $e_1 \wedge e_2$ and called the **meet** of e_1 and e_2. If every pair of elements in $\mathcal{P}$ has a g.l.b. and an l.u.b. we say

that $(\mathcal{P}, \leq)$ is a **lattice**. A **lattice** can be defined independently of a partial order taking two operations '$\vee$' and '$\wedge$' as primitives satisfying the properties given below:
(idempotency) $x \vee x = x \quad \forall\, x \in \mathcal{P}$; $x \wedge x = x \quad \forall\, x \in \mathcal{P}$.
(commutativity) $x \vee y = y \vee x \quad \forall\, x, y \in \mathcal{P}$; $x \wedge y = y \wedge x \quad \forall\, x, y \in \mathcal{P}$.
(associativity) $(x \vee y) \vee z = x \vee (y \vee z) \quad \forall\, x, y, z \in \mathcal{P}$.
$(x \wedge y) \wedge z = x \wedge (y \wedge z) \quad \forall\, x, y, z \in \mathcal{P}$.
(absorption) $x \wedge (x \vee y) = x \vee (x \wedge y) = x \quad \forall\, x, y \in \mathcal{P}$.
The reader may verify that these properties are indeed satisfied by g.l.b. and l.u.b. operations if we start from a partial order.
A lattice that satisfies the following additional property is called a **distributive lattice**.
(distributivity) $x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z) \quad \forall\, x, y, z \in \mathcal{P}$;
$x \vee (y \wedge z) = (x \vee y) \wedge (x \vee z) \quad \forall\, x, y, z \in \mathcal{P}$.
e.g. The collection of all subsets of a given set with union as the join operation and intersection as the meet operation is a distributive lattice. (For a comprehensive treatment of lattice theory see [Birkhoff67]).

Exercise 2.66 *Show that the collection of ideals of a partial order form a distributive lattice under union and intersection.*

2.6.8 Partitions

Let S be a finite set. A collection $\{S_1, \cdots, S_k\}$ of nonvoid subsets of S is a **partition** of S iff $\bigcup_i S_i = S$ and $S_i \cap S_j = \emptyset$ whenever i, j are distinct. If $\{S_1, \cdots, S_k\}$ is a partition of S then the S_i are referred to as its **blocks**.
Let $\mathcal{P}_S$ denote the collection of all partitions of S. We may define a partial order $(\mathcal{P}, \leq)$ on $\mathcal{P}_S$ as follows: Let $\Pi_1, \Pi_2 \in \mathcal{P}_S$. Then $\Pi_1 \leq \Pi_2$ (equivalently $\Pi_2 \geq \Pi_1$) iff each block of Π_1 is contained in a block of Π_2. We say Π_1 is **finer than** Π_2 or Π_2 is **coarser than** Π_1. If Π_a, Π_b are two partitions of S, the **join** of Π_a and Π_b, denoted by $\Pi_a \vee \Pi_b$, is the finest partition of S that is coarser than both Π_a and Π_b and the **meet** of Π_a and Π_b denoted by $\Pi_a \wedge \Pi_b$, is the coarsest partition of S that is finer than both Π_a and Π_b. It can be seen that these notions are well defined: To obtain the meet of Π_a and Π_b we take the intersection of each block of Π_a with each block of Π_b and throw away the empty intersections. Observe that any element of S lies in precisely one such

intersection. Clearly, the resulting partition Π_{ab} is finer than both Π_a and Π_b. Suppose Π_c is finer than both Π_a and Π_b. Let N_c be a block of Π_c. Then N_c is contained in some block N_a of Π_a and some block N_b of Π_b. So $N_c \subseteq N_a \cap N_b$ and hence N_c is contained in some block of Π_{ab}. This proves that Π_{ab} is the meet of Π_a and Π_b and therefore, that the 'meet' is well defined. Next let Π, Π' be two partitions coarser than Π_a and Π_b. It is then easy to see that $\Pi \wedge \Pi'$ is also coarser than Π_a and Π_b. Hence there is a unique finest partition of S coarser than Π_a and Π_b. Thus, the 'join' is well defined.

Storing partitions: We can store a partition by marking against an element of S, the name of the block to which it belongs.

Building $\Pi_a \wedge \Pi_b$: When Π_a, Π_b are stored, each element of S would have against it two names - a block of Π_a and a block of Π_b; a pair of names of intersecting blocks of Π_a, Π_b can be taken to be the name of a block of $\Pi_a \wedge \Pi_b$. Thus forming $\Pi_a \wedge \Pi_b$ from Π_a, Π_b is $O(\mid S \mid)$.

Building $\Pi_a \vee \Pi_b$: We first build a bipartite graph B with blocks of Π_a as V_L, blocks of Π_b as V_R with an edge between $N_a \in V_L$ and $N_b \in V_R$ iff $N_a \cap N_b \neq \emptyset$. It can be seen that this bipartite graph can be built in $O(\mid S \mid)$ time (For each element of S, check which blocks of Π_a, Π_b it belongs to). We find the connected components of this bipartite graph. This can be done in $O(m + n)$ time where m is the number of edges and n, the number of vertices in the bipartite graph. But both m and n do not exceed $\mid S \mid$. So $O(m + n) = O(\mid S \mid)$. Now we collect blocks of Π_a (or Π_b) belonging to the same connected component of B. Their union would make up a block of $\Pi_a \vee \Pi_b$. (For, this block is a union of some blocks K of Π_a as well as a union of some blocks of Π_b. Union of any proper subset of blocks of K would cut some block of Π_b). This involves changing the name marked against an element $u \in S$ - instead of say N_a, it would be N_v, which is the name of the connected component of B in which N_a is a vertex. Thus, building $\Pi_a \vee \Pi_b$ is $O(\mid S \mid)$.

2.6.9 The Max-Flow Problem

In this subsection we outline the max-flow problem and a simple solution for it. We also indicate the directions in which more sophisticated solutions lie. In subsequent chapters we use max-flow repeatedly to model various minimization problems. Other than the flexibility in modeling that it offers, the practical advantage of using the concept of max-flow lies in the availability of efficient algorithms.

Let $\mathcal{G}$ be a directed graph. The **flow graph** (or flow network) $F(\mathcal{G})$ is the tuple $(\mathcal{G}, \mathbf{c}$, s,t) where $c : E(\mathcal{G}) \rightarrow \Re_+$ is a real nonnegative **capacity** function on the edges of $\mathcal{G}$ and s and t are two vertices of $\mathcal{G}$ named **source** and **sink**, respectively. A **flow f** associated with $F(\mathcal{G})$ is a vector on $E(\mathcal{G})$ satisfying the following conditions:

i. **f** satisfies KCE at all nodes except s and t, i.e., at each vertex v other than s, t, the **net outward flow**

$$\sum_i f(e_{outi}) - \sum_j f(e_{inj}) = 0$$

where $e_{outi}(e_{inj})$ are the edges incident at v and directed out of (directed into) v.

ii. the net outward flow at s is nonnegative, and at t, is non-positive.

iii. $0 \leq f(e) \leq c(e) \quad \forall\ e \in E(\mathcal{G})$.

(Often a flow is defined to be a vector satisfying (i) and (ii) above while a **feasible flow** would be one that satisfies all three conditions). An edge e with $f(e) = c(e)$ is said to be **saturated** with respect to **f**. The **value** of the flow **f**, denoted by $\mid \mathbf{f} \mid$, is the net outward flow at s. A flow of maximum value is called a **max-flow**. An **s,t-cut** (**cut** for short) is an ordered pair (A, B), where A, B are disjoint complementary subsets of $V(\mathcal{G})$ s.t. $s \in A$ and $t \in B$. The **capacity** of the cut (A, B), denoted by $c(A, B)$ is the sum of the capacities of edges with positive end in A and negative end in B. A cut of minimum capacity is called a **min-cut**. The **flow across** (A, B) denoted by $f(A, B)$, is the sum of the flows in the 'forward' edges going from A to B minus the sum of the flows in the 'backward' edges going from B to A.

Example: Figure 2.12 shows a flow graph. Alongside each directed edge is an ordered pair with the second component indicating the capacity of the edge. A feasible flow **f** is defined on this flow graph with $f(e)$ being the first component of the ordered pair alongside e. The reader may verify that the net flow leaving any node other than the

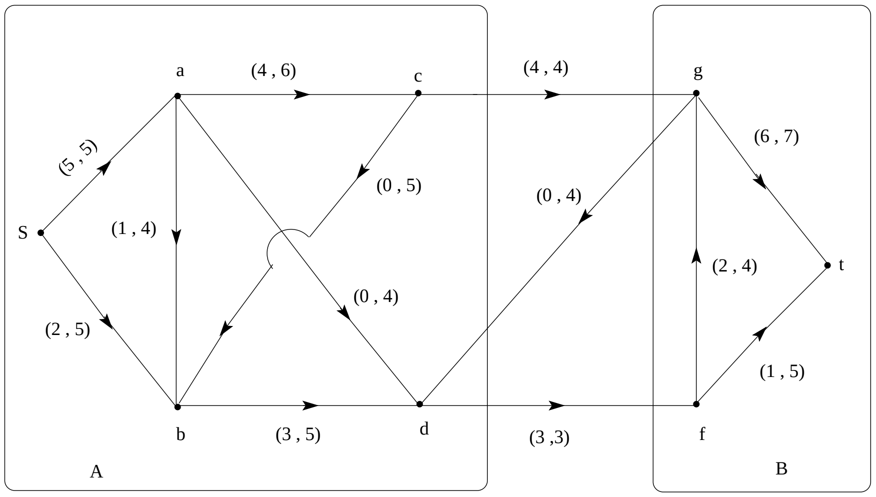


Figure 2.12: A Flow Graph with a max-flow and a min-cut

source s and the sink t is zero. At s there is a net positive outward flow ($= 7$) and at t there is a net negative outward flow ($= -7$). Let $A \equiv \{s, a, b, c, d\}$ and let $B \equiv \{g, f, t\}$. Then (A, B) is an s, t cut. It can be verified that $f(A, B) = 4+3-0 = 7$. Observe that the forward edges (c, g) and (d, f) of the cut (A, B) are saturated while the backward edge (g, d) carries zero flow. It is clear that in the present case $f(A, B) = c(A, B)$. From the arguments given below it would follow that the given flow has the maximum value, i.e, is a max-flow and that the cut (A, B) is a min-cut, i.e., has minimum capacity.

Clearly the flow across an s,t-cut (A, B) cannot exceed the capacity of (A, B), i.e., $f(A, B) \leq c(A, B)$. Let (A, B) be an s,t-cut. If we add the outward flows at all nodes inside A we would get the value $f(A, B)$ (flow of each edge with both ends within A is added once with a $(+)$ sign and another time with a $(-)$ sign and hence cancels) as well as $\mid \mathbf{f} \mid$ (at all nodes other than s the net outward flow is zero). We conclude that $\mid \mathbf{f} \mid = f(A, B)$.

Let $\mathbf{f}$ be a flow in $F(\mathcal{G})$. Let P be a path oriented from s to t. Suppose it is possible to change the flow in the edges of P, without violating capacity constraints, as follows: the flow in each edge e of P is increased (decreased) by $\delta > 0$ if e supports (opposes) the orientation of P.

Such a path is called an **augmenting** path for the flow $\mathbf{f}$. Observe that this process does not disturb the KCE at any node except s, t. At s, the net outward flow goes up by δ, while at t, the net inward flow goes up by δ. Thus, if $\mathbf{f}'$ is the modified flow, $\mid \mathbf{f}' \mid = \mid \mathbf{f} \mid + \delta$. This is the essential idea behind flow maximization algorithms.

It is convenient to describe max-flow algorithms and related results in terms of the **residual graph** $\mathcal{G}_f$ associated with the flow $\mathbf{f}$. The graph $\mathcal{G}_f$ has the vertex set $V(\mathcal{G})$. Whenever $e \in E(\mathcal{G})$ and $f(e) < c(e)$, $\mathcal{G}_f$ has an edge e_+ between the same end points and in the same direction as e; and if $0 < f(e)$, $\mathcal{G}_f$ has an edge e_- in the opposite direction as e. Note that both e_+ and e_- may be present in $\mathcal{G}_f$. The edge e_+ has the **residual capacity** $r_f(e_+) \equiv c(e) - f(e)$ and the edge e_- has the **residual capacity** $r_f(e_-) \equiv f(e)$.
We note that a directed path P from s to t in the residual graph $\mathcal{G}_f$ corresponds to an augmenting path in $F(\mathcal{G})$ with respect to $\mathbf{f}$. Henceforth we would call such a path P in $\mathcal{G}_f$ also, an **augmenting path** of $\mathbf{f}$. The maximum amount by which the flow can be increased using this augmenting path is clearly the minimum of the residual capacities of the edges of P. This value we would call the **bottle neck capacity** of P.

We now present a simple algorithm for flow maximization. This algorithm is due to Edmonds and Karp [Edmonds+Karp72].

ALGORITHM 2.1 Algorithm Max-Flow
INPUT *A flow graph* $F(\mathcal{G}) \equiv (\mathcal{G}, \mathbf{c}, s, t)$.

OUTPUT *(i) A maximum valued flow* $\mathbf{f}_{max}$ *for* $F(\mathcal{G})$.
(ii) A min-cut (A, B) *s.t.* $\mid \mathbf{f}_{max} \mid \equiv c(A, B)$.

Initialize *Let* $\mathbf{f}$ *be any flow of* $F(\mathcal{G})$ *(* $\mathbf{f}$ *could be the zero flow for instance).*

STEP 1 *Draw the residual graph* $\mathcal{G}_f$. *Do a directed* bfs *starting from* s.

STEP 2 **If t is reached,** *we also have a shortest augmenting path P. Compute the bottle neck capacity δ of P. Increase the flow along P by δ. Let $\mathbf{f}'$ be the new flow. Set $\mathbf{f} \equiv \mathbf{f}'$ and GOTO STEP 1.*

If t is not reached, *let A be the set of all vertices reached from s and let $B \equiv V(\mathcal{G}) - A$. Declare $\mathbf{f}_{max} \equiv \mathbf{f}$, min-cut to be (A, B).*

STOP.

Justification of Algorithm 2.1

We need the following

Theorem 2.6.1 *(***Max-Flow Min-Cut Theorem** *[Ford+Fulkerson56], [Ford+Fulkerson62])*

i. The flow reaches its maximum value iff there exists no augmenting path.

ii. The maximum value of a flow in $F(\mathcal{G})$ is the minimum value of the capacity of a cut.

Proof : If a flow has maximum value it clearly cannot permit the existence of an augmenting path. If there exists no augmenting path the directed bfs from s in the residual graph will not reach t. Let A be the set of all vertices reached from s and let B be the complement of A. All edges with one end in A and the other in B must be directed into A as otherwise the set of reachable vertices can be enlarged. Now consider the corresponding edges of $F(\mathcal{G})$. Each one of these edges, **if forward** (away from A), must have reached full capacity, i.e., be saturated and **if backward** (into A), must have zero flow. But then, for this cut, $f(A, B) = c(A, B)$. Since for any flow $\mathbf{f}$ and any cut (A', B'), we have $\mid f \mid = f(A', B') \leq c(A', B')$, we conclude that $\mathbf{f}$ is a maximum flow. This completes the proof of (i).
Since $\mid f \mid = c(A, B)$ and $\mid f \mid \leq c(A', B')$ for any cut (A', B'), it is clear that $c(A, B)$ is the minimum capacity of a cut of $F(\mathcal{G})$. This proves (ii).

□

The integral capacity case: We can justify the above algorithm for the case where capacities are **integral** quite simply. Let M be the capacity of the cut $(\{s\}, V(\mathcal{G}) - s)$. The bottle neck capacity of any augmenting path is integral. Whenever we find an augmenting path we would increase the flow by an integer and Theorem 2.6.1 assures us that if we are unable to find an augmenting path we have reached max-flow. Thus, in atmost M augmentations we reach maximum flow. This justification also proves the following corollary.

Corollary 2.6.1 *If the capacity function of a flow graph is integral, then there exists a max-flow in the flow graph which is integral.*

Complexity

We consider the integral capacity case. Each augmentation involves a directed bfs. This is $O(m)$ in the present case. Hence, the overall complexity of **Algorithm Max-Flow** is $O(Mm)$, where $m \equiv\mid E(\mathcal{G}) \mid$.

It is not obvious that **Algorithm Max-Flow** would terminate for real capacities. However, it can be shown that it does. Since the augmenting path is constructed through a bfs it is clear that it has minimum length. Edmonds and Karp [Edmonds+Karp72] have shown that if the shortest augmenting path is chosen every time, there are atmost mn augmentations. So the overall complexity of **Algorithm Max-Flow** is $O(m^2 n)$.

Exercise 2.67 *[Edmonds+Karp72] In Algorithm Max-Flow, if the shortest augmenting path is chosen every time, show that there are atmost mn augmentations.*

We mention a few other algorithms which are faster. These are based on Dinic's Algorithm [Dinic70]. This algorithm proceeds in **phases**, in each of which, flow is pushed along a maximal set of shortest paths. Each phase takes $O(mn)$ effort. The total number of phases is bounded by the length L of the longest $s-t$ path in $\mathcal{G}$ (Clearly $L \leq n$). So the overall complexity is $O(Lmn)$.
The MPM Algorithm [MPM78] has the same number of phases as Dinic's Algorithm. But each phase is $O(n^2)$. So the overall complexity is $O(Ln^2)$.

The Sleator Algorithm [Sleator80], [Sleator+Tarjan 83] computes each phase in $O(m \log n)$ time and has an overall complexity $O(Lm \log n)$. (Usually the above complexities are stated with n in place of L). For a comprehensive treatment of flow algorithms the reader is referred to [Ahuja+Magnanti+Orlin93].

The Nearest Source Side and Sink Side Min-Cuts

When combinatorial problems are modelled as max-flow problems, usually the cuts with minimum capacity have physical significance. Of particular interest would be minimum capacity cuts (A, B) where A or B is minimal. Below we show that these cuts are unique. Further, we show that computing them, after a max-flow has been found, is easy. We begin with a simple lemma.

Lemma 2.6.1 *(k) Let* $(A_1, B_1), (A_2, B_2)$ *be two minimum capacity cuts. Then* $(A_1 \cup A_2, B_1 \cap B_2)$ *and* $(A_1 \cap A_2, B_1 \cup B_2)$ *are also minimum capacity cuts.*

Proof : Let $f(A) \equiv$ sum of the capacity of edges with one end in A and directed away from $A, A \subseteq V(\mathcal{G})$.
Later, in Chapter **??** (see Exercise **??**, Examples **??**,**??**) we show that $f(\cdot)$ is submodular, i.e.,

$$f(X) + f(Y) \geq f(X \cup Y) + f(X \cap Y) \ \forall \ X, Y \subseteq V(\mathcal{G}).$$

Now if X, Y minimize $f(\cdot)$, the only way the above inequality can be satisfied is for $f(\cdot)$ to take the minimum value on $X \cup Y, X \cap Y$ also. This proves the lemma.

□

The following corollary is now immediate.

Corollary 2.6.2 *Let* $F(\mathcal{G}) \equiv (\mathcal{G}, \mathbf{c}, s, t)$. *Then* $F(\mathcal{G})$ *has a unique min-cut* (A, B) *in which* A *is minimal* (B *is minimal).*

We will call the min-cut (A, B) **nearest source side (sink side) min-cut** iff A is minimal (B is minimal). To find the nearest source side (sink side) min cut we proceed as follows
Algorithm Source (Sink) Side Min-Cut: First maximize flow and let $\mathbf{f}$ be the max-flow output by the algorithm. Draw the residual graph

$\mathcal{G}_f$. Do a directed bfs in $\mathcal{G}_f$ starting from s and proceeding forward. Let A_s be the set of all vertices reachable from s. Then $(A_s, V(\mathcal{G})-A_s)$ is the desired nearest source side min-cut.
Let $\mathcal{G}_f^-$ denote the directed graph obtained from $\mathcal{G}_f$ by reversing all arrows. The nearest sink side min-cut is obtained by doing a directed bfs starting from t in $\mathcal{G}_f^-$. Let B_t be the set of all vertices reachable in $\mathcal{G}_f^-$ from t. Then $(V(\mathcal{G}) - B_t, B_t)$ is the desired nearest sink side min-cut.

In order to justify the above algorithms we first observe that when we maximize flow for each min-cut (A, B) we would have $f(A, B) = c(A, B)$. Thus, if (A, B) is a min-cut, all the forward edges from A to B would be saturated and all the backward edges from A to B would have zero flow. Therefore, in the residual graph $\mathcal{G}_f$ all edges across the cut would be directed into A. Now $s \in A$ and doing a bfs starting from s we cannot go outside A. Hence, if (A, B) is a min-cut $A_s \subseteq A$, where A_s is the set of all vertices reachable from s in $\mathcal{G}_f$. But $(A_s, V(\mathcal{G})-A_s)$ is a min-cut. Hence, $(A_s, V(\mathcal{G})-A_s)$ is the nearest source side min-cut. The justification for the sink side min-cut algorithm is similar. (Note that the above justification provides an alternative proof that min-cuts (A, B), where A or B is minimal, are unique).

The complexity of the above algorithms is $O(m)$. So if they are added to the max-flow algorithms the overall complexity would not increase.

2.6.10 Flow Graphs Associated with Bipartite Graphs

Many optimization problems considered in this book are based on bipartite graphs. Usually they reduce to max- flow problems on a flow graph derived from the bipartite graph in a simple manner. We give below a brief account of the situation and standardize notation.

Let $B \equiv (V_L, V_R, E)$ be a bipartite graph. The flow graph $F(B, \mathbf{c}_L, \mathbf{c}_R)$ **associated with** B **with capacity** $c_L(\cdot) \oplus c_R(\cdot)$ is defined as follows:

$c_L(\cdot), c_R(\cdot)$ are nonnegative real functions on V_L, V_R respectively. (They may therefore be treated as weight vectors). Each edge $e \in E$ is directed from V_L to V_R and given a capacity ∞. Additional vertices (source) s and (sink) t are introduced. Directed edges $(s, v_L), (v_R, t)$ are added for each $v_L \in V_L$ and each $v_R \in V_R$. The capacity of the edge (s, v_L) is $c_L(v_L), v_L \in V_L$ and the capacity of the edge (v_R, t) is $c_R(v_R), v_R \in V_R$. Figure 2.13 illustrates the construction of this flow graph.

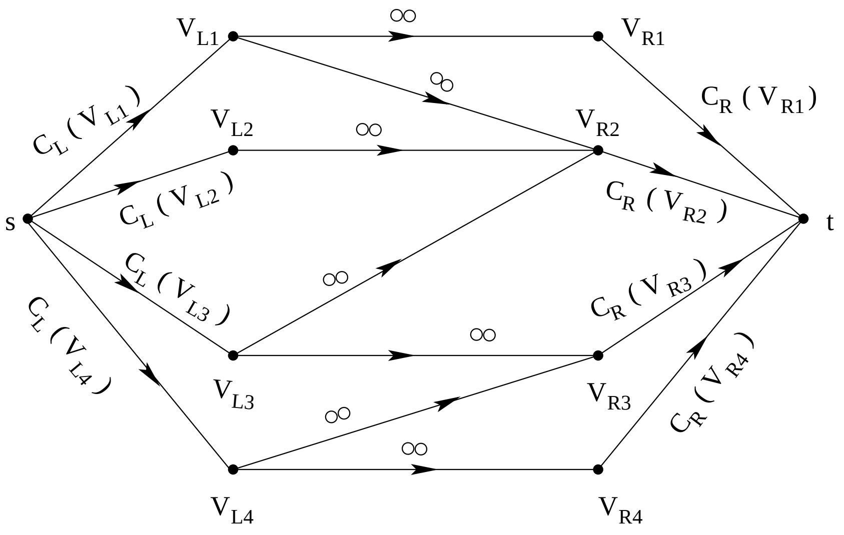


Figure 2.13: The Flow Graph associated with B

Let $\Gamma(X)$ denote the set of vertices adjacent to the vertex subset $X \subseteq V_L \uplus V_R$ in the bipartite graph B. Let $c_L(Z)$ ($c_R(Z)$) denote the sum of the values of $c_L(\cdot)$ ($c_R(\cdot)$) on elements of Z. The **cut corresponding to** $X \subseteq V_L$ is the cut $(s \uplus X \uplus \Gamma(X), t \uplus (V_L - X) \uplus (V_R - \Gamma(X)))$ (see Figure 2.14). We now have the following simple theorem which brings out the utility of the flow formulation.

Theorem 2.6.2 *(k) Let $B, c_L(\cdot), c_R(\cdot), F(B, \mathbf{c}_L, \mathbf{c}_R)$ be as defined above.*

i. The capacity of the cut corresponding to X, $X \subseteq V_L$, is $c_L(V_L - X) + c_R(\Gamma(X))$.

ii. $Z \subseteq V_L$ minimizes the expression $c_L(V_L - X) + c_R(\Gamma(X)), X \subseteq V_L$, iff the cut corresponding to Z is a min-cut of $F(B, \mathbf{c}_L, \mathbf{c}_R)$.

iii. There is a unique maximal subset Z_{max} and a unique minimal set Z_{min} which minimize the above expression. Let $c_R(\cdot)$ be strictly

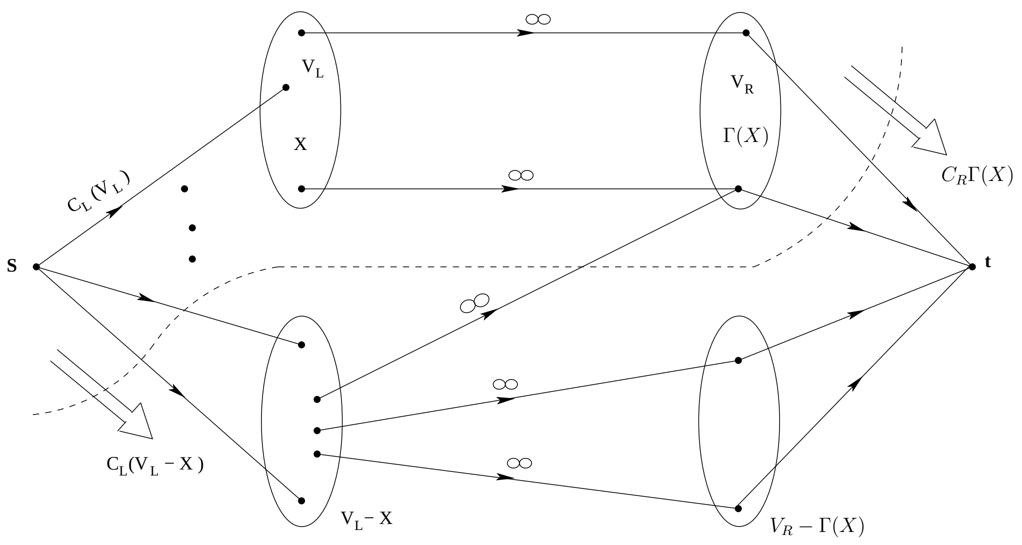


Figure 2.14: The Cut corresponding to X in the Flow Graph associated with B

positive. Then the cuts corresponding to Z_{max}, Z_{min} are respectively the nearest sink side and the nearest source side min-cuts of $F(B, \mathbf{c}_L, \mathbf{c}_R)$.

Proof :
i. This is immediate (see Figure 2.14).

ii. We will first show that there exist min-cuts which are cuts corresponding to some $X_1 \subseteq V_L$.

Let $(s \uplus X_1 \uplus Y, t \uplus (V_L - X_1) \uplus (V_R - Y))$ be a min-cut of $F(B, \mathbf{c}_L, \mathbf{c}_R)$, where $X_1 \subseteq V_L, Y \subseteq V_R$. Since this is a min-cut, no infinite capacity edge must pass from $s \uplus X_1 \uplus Y$ to its complement. This means that any edge leaving X_1 must terminate in Y, i.e., $\Gamma(X_1) \subseteq Y$. The capacity of the cut is $c_L(V_L - X_1) + c_R(Y)$. Now consider the cut $(s \uplus X_1 \uplus \Gamma(X_1), t \uplus (V_L - X_1) \uplus (V_R - \Gamma(X_1)))$. The capacity of this cut is $c_L(V_L - X_1) + c_R(\Gamma(X_1)) \leq c_L(V_L - X_1) + c_R(Y)$, ($\mathbf{c}_L, \mathbf{c}_R$ are nonnegative vectors). Thus the cut corresponding to X_1, is a min cut.
Let Z minimize the expression $c_L(V_L - X) + c_R(\Gamma(X)), X \subseteq V_L$, and let the cut corresponding to $Z' \subseteq V_L$ be a min-cut of $F(B, \mathbf{c}_L, \mathbf{c}_R)$. The capacity of this cut is $c_L(V_L - Z') + c_R(\Gamma(Z'))$. So $c_L(V_L - Z) + c_R(\Gamma(Z)) \leq c_L(V_L - Z') + c_R(\Gamma(Z'))$. However the LHS is the capacity

of the cut corresponding to Z. Since the cut corresponding to Z' is a min cut we must have $c_L(V_L - Z) + c_R(\Gamma(Z))$
$\geq c_L(V_L - Z') + c_R(\Gamma(Z'))$. We conclude that the two capacities are equal. Therefore Z' minimizes $c_L(V_L - X) + c_R(\Gamma(X)), X \subseteq V_L$, and the cut corresponding to Z is a min-cut.

iii. The nearest source side min-cut can be seen to be corresponding to some subset X_1 of V_L even if $c_R(\cdot)$ is nonnegative, but, not necessarily, strictly positive.
Now let $c_R(\cdot)$ be strictly positive.
The nearest sink side min-cut is obtained by travelling backward from t through all unsaturated arcs to reach $T_R \subseteq V_R$ and then backwards to $\Gamma(T_R) \subseteq V_L$. The cut that we obtain by this process is $(s \uplus X_2 \uplus (V_R - T_R), t \uplus \Gamma(T_R) \uplus T_R)$ where $X_2 \equiv V_L - \Gamma(T_R)$. It is clear that $\Gamma(X_2) \subseteq V_R - T_R$. Suppose $\Gamma(X_2) \subset V_R - T_R$. Then the capacity of the cut corresponding to $X_2 = c_L(V_L - X_2) + c_R(\Gamma(X_2))$
$< c_L(V_L - X_2) + c_R(V_R - T_R)$, since c_R is strictly positive. The RHS of the above inequality is the capacity of the min-cut $(s \uplus X_2 \uplus (V_R - T_R), t \uplus \Gamma(T_R) \uplus T_R)$ - a contradiction. We conclude that $\Gamma(X_2) = V_R - T_R$, so that the nearest sink side min-cut corresponds to X_2.

We know that X_1, X_2 minimize the expression $c_L(V_L - X) + c_R(\Gamma(X))$, $X \subseteq V_L$. Let $A \subset s \uplus X_1 \uplus \Gamma(X_1)$ and let B be the complement of A with respect to $V_L \uplus V_R \uplus \{s, t\}$. Then (A, B) cannot be a min-cut (using the justification for the Algorithm Source Side Min Cut). Also $s \uplus X_1 \uplus \Gamma(X_1)$ is the unique set with the above property. It follows that X_1 is the unique minimal set s.t.
$(s \uplus X_1 \uplus \Gamma(X_1), t \uplus (V_L - X_1) \uplus (V_R - \Gamma(X_1)))$ is a min-cut. Hence, X_1 is the minimal set that minimizes $c_L(V_L - X) + c_R(\Gamma(X))$. The proof that X_2 is the maximal set that minimizes the above expression is similar.

$\square$

Remark: The expression that was minimized in the above proof is a submodular function. We shall see later, in Chapter **??**, that such functions always have a unique minimal and a unique maximal set minimizing them.

Complexity of Max-Flow Algorithms for Bipartite Graph Case

Finally we make some observations on the complexity of the max flow algorithms when the flow graph is associated with a bipartite graph. We note that, in this case, the **longest undirected path from** s **to** t is $O(\min(| V_L |, | V_R |))$, since every path from s to t has to alternate between vertices of V_L, V_R. So the **number of phases** for Dinic's (and related) algorithms would be $O(\min(| V_L |, | V_R |))$. Therefore the overall complexities of the algorithms for this case would be

$$\begin{array}{rcl} Dinic's & - & O(mn(\min(| V_L |, | V_R |))) \\ MPM & - & O(n^2(\min(| V_L |, | V_R |))) \\ Sleator & - & O(m \log n(\min(| V_L |, | V_R |))). \end{array}$$

Here, n, m refer to the total number of vertices and edges respectively in the flow graph. So
$n =| V_L | + | V_R | +2$ and $m =| E | + | V_L | + | V_R |$.

Exercise 2.68 *[Menger27] In any graph, show that the number of arc disjoint paths, between any pair of vertices* s *and* t*, is the number of branches in a min-cut separating* s *and* t*.*

2.7 Duality

Duality is a useful concept often met with in mathematics, e.g. duality of vector spaces and spaces of functionals, duality of partial orders, duality of functions and Fourier transforms etc. When we encounter it we need to know **why** it arises and **how** to use it. The duality that one normally deals with in electrical network theory, arises because the voltage and current spaces of graphs are complementary orthogonal. (For other examples of duality that one encounters within electrical network theory see [Iri+Recski80]). In this section we discuss informally **how to dualize** statements about graphs, vector spaces (and therefore, implicitly, electrical networks) and also as to when we may expect the dual of a true statement to be true.

Let $\mathcal{V}$ be a vector space on S. We associate with $\mathcal{V}$

i. a set of operations each of which converts $\mathcal{V}$ to a vector space on a subset of S - a typical operation is $(S-T_1, T_1-T_2)(\cdot), T_2 \subseteq T_1 \subseteq S$, where

$$(S-T_1, T_1-T_2)(\mathcal{V}) \equiv \mathcal{V} \ . \ T_1 \times T_2;$$

ii. classes of objects:

- class of forests
- class of coforests
- class of circuits
- class of cutsets
- primal vectors (vectors in $\mathcal{V}$)
- dual vectors (vectors in $\mathcal{V}^{\perp}$).

Remark: For convenience we generalize the usual definitions of forest, coforest, circuit, cutset etc. to vector spaces. The reader may verify that, if $\mathcal{V}$ were replaced by $\mathcal{V}_v(\mathcal{G})$, these definitions do reduce to the usual definitions in terms of graphs. A **forest** of $\mathcal{V}$ is a maximally independent subset of columns of a representative matrix of $\mathcal{V}$, a **coforest** of $\mathcal{V}$ is the complement, relative to the set of columns of the representative matrix, of a forest, a **circuit** of $\mathcal{V}$ is a minimal set that is not contained in any forest of $\mathcal{V}$, while a **cutset** of $\mathcal{V}$ is a minimal set that is not contained in any coforest of $\mathcal{V}$. The classes of coforests, circuits and cutsets are used for convenience. Actually any one of the four classes can be treated as primitive and the rest expressed in terms of it.

Now we list the results which 'cause' duality.

i. $(\mathcal{V}^{\perp})^{\perp} = \mathcal{V}$, equivalently, $\mathbf{x}$ is a primal vector for $\mathcal{V}$ iff it is a dual vector for $\mathcal{V}^{\perp}$.

ii. $(\mathcal{V} \ . \ T_1 \times T_2)^{\perp} = \mathcal{V}^{\perp} \times T_1 \cdot T_2 = \mathcal{V}^{\perp} \cdot (S-(T_1-T_2)) \times T_2$, i.e., the operation $(S-T_1, T_1-T_2)(\cdot)$ holds the same place relative to $\mathcal{V}$, that the operation $(T_1-T_2, S-T_1)(\cdot)$ holds, relative to $\mathcal{V}^{\perp}$. We say $(S-T_1, T_1-T_2)(\cdot)$ is **dual** to $(T_1-T_2, S-T_1)(\cdot)$.

iii. (later we add one more operation which includes all the above, namely, that of generalized minor)

$$(\mathcal{V}_S \leftrightarrow \mathcal{V}_P)^{\perp} = \mathcal{V}_S^{\perp} \leftrightarrow \mathcal{V}_P^{\perp}, \quad P \subseteq S.$$

iv. T is a forest (coforest) of $\mathcal{V}$ iff T is a coforest (forest) of $\mathcal{V}^{\perp}$.

v. C is a circuit (cutset) of $\mathcal{V}$ iff C is a cutset (circuit) of $\mathcal{V}^{\perp}$.

Let us consider how to '**dualize**' a statement about a vector space and the associated set of operations and classes of objects. Our procedure requires that the statement to be dualized be in terms of the primitive objects and operations, associated with a vector space, that we described above. Consider the statement
i. 'A subset is a circuit of $\mathcal{V} \times T$ iff it is a minimal intersection of a circuit of $\mathcal{V}$ with T '.
The first step is to write the statement in terms of $\mathcal{V}^{\perp}$:
'A subset is a circuit of $\mathcal{V}^{\perp} \times T$ iff it is a minimal intersection of a circuit of $\mathcal{V}^{\perp}$ with T '.
Next we try to express the sets of objects involved in terms of the appropriate complementary orthogonal space. Thus 'circuit of $\mathcal{V}^{\perp} \times T$ ' becomes 'cutset of $(\mathcal{V}^{\perp} \times T)^{\perp}$ ' and 'circuit of $\mathcal{V}^{\perp}$ ' becomes 'cutset of $(\mathcal{V}^{\perp})^{\perp}$ ' we thus obtain the dual of (i):
i^d. 'A subset is a cutset of $\mathcal{V}$. T iff it is a minimal intersection of a cutset of $\mathcal{V}$ with T'.

The above procedure will yield a true (false) dual statement if we start with a true (false) statement. However, as we mentioned before, the statement that we start with must involve only the 'primitives' viz. the sets of operations and the classes of objects.

Next let us consider the case of (directed) graphs. We associate with a graph, a vector space, namely, its voltage space. Given a statement about graphs we first see whether it can be written entirely in terms of its voltage space. If so, then we dualize it and interpret the dual statement in terms of graphs. For instance consider the statement
ii. 'A subset is a circuit of $\mathcal{G} \times T$ iff it is a minimal intersection of a circuit of $\mathcal{G}$ with T '.
This statement can be written entirely in terms of $\mathcal{V}_v(\mathcal{G})$. If we substitute $\mathcal{V}$ in place of $\mathcal{V}_v(\mathcal{G})$ in this latter statement, we get the statement (i) above. Its dual is (i^d). Now we resubstitute $\mathcal{V}_v(\mathcal{G})$ in place of $\mathcal{V}$. This gives us
'A subset is a cutset of $\mathcal{V}_v(\mathcal{G}) \cdot T$ iff it is a minimal intersection of a cutset of $\mathcal{V}_v(\mathcal{G})$ with T '.
Interpreting this statement in terms of $\mathcal{G}$ gives us

ii^d. ' A subset is a cutset of $\mathcal{G} \cdot T$ iff it is a minimal intersection of a cutset of $\mathcal{G}$ with T '.
The above procedure could fail in the beginning when we try to write the statement about $\mathcal{G}$ as a statement about $\mathcal{V}_v(\mathcal{G})$ or when we replace $\mathcal{V}_v(\mathcal{G})$ by a general $\mathcal{V}$ (all $\mathcal{V}_v(\mathcal{G})$ satisfy properties that all $\mathcal{V}$ do not). It could also fail when we replace $\mathcal{V}$ by $\mathcal{V}_v(\mathcal{G})$ in the dual statement.

Here are a couple of examples of statements which cannot be dualized by our procedure.

i. 'Let $\mathcal{G}$ be a connected graph and let f be a forest of $\mathcal{G}$. Then there exists a unique path between any given pair of vertices using the edges of f alone '.
The procedure fails because 'path' and 'vertices' cannot be extended to vector spaces.

ii. 'There exists a graph $\mathcal{G}$ that has the given sets of edges $C_1, \cdots, C_n$ as circuits'.
We can extend this to $\mathcal{V}_v(\mathcal{G})$, thence to $\mathcal{V}$, and dualize the statement involving $\mathcal{V}$. This statement would be:
'There exists a vector space $\mathcal{V}$ that has the given sets of edges $C_1, \cdots, C_n$ as cutsets'.
The procedure can fail if we replace $\mathcal{V}$ by $\mathcal{V}_v(\mathcal{G})$ since the latter statement may be false.

Exercise 2.69 *What are the duals of the following?*

i. rank function of a graph

ii. $r(\cdot)$ *where* $r(T) \equiv dim(\mathcal{V} \cdot T)$

iii. $\xi(\cdot)$, *where* $\xi(T) \equiv dim(\mathcal{V} \cdot T) - dim(\mathcal{V} \times T)$

iv. Closed sets of a graph (a subset of edges is closed if its rank is less than that of any proper superset)

v. selfloops

vi. coloops

vii. separators of $\mathcal{V}$

viii. separators (2 connected components) of a graph.

Exercise 2.70 *Dualize the following statements. Assuming the truth of the original statement, comment on the truth of the dual.*

i. *A coforest is a minimal set that intersects every circuit.*

ii. *A circuit is a minimal set that intersects every coforest.*

iii. *Every ring sum of circuits of $\mathcal{G}$ is a disjoint union of circuits. ($C_1 +_r \cdots +_r C_n$ is the set of all elements which occur in an odd number of the C_i).*

iv. *Let C_1, C_2 be circuits of $\mathcal{G}$ and let $e_c \in C_1 \cap C_2$ and $e_1 \in C_1 - C_2$. Then there exists a circuit C_3 of $\mathcal{G}$ s.t. $e_1 \in C_3 \subseteq C_1 \cup C_2 - e_c$.*

v. *Let $\mathcal{G}$ be a graph and let $E(\mathcal{G})$ be partitioned into $E_1, \cdots, E_n$. Let f be a forest of $\mathcal{G}$ which has as many edges as possible of E_1, then as many as possible of $E_2 \cdots$ upto E_n. Then $f \cap E_j$ is a forest of $\mathcal{G} \ . \ (\bigcup_{i=1}^{j} E_i) \times E_j, j = 1, \cdots, k$.*

vi. *Let $\mathcal{G}$ be a graph. Let $E \equiv E(\mathcal{G})$ be partitioned into sets A, B. Then $L \subseteq B$ is a minimal set such that $\mathcal{G} \ . \ (E - L)$ has A as a separator iff*

(a) $r(\mathcal{G} \times (A \cup L)) = r(\mathcal{G} \ . \ A)$.

(b) *L has no self loops in $\mathcal{G} \times (A \cup L)$.*

vii. *Let $\mathcal{V}$ be a vector space on S and let S be partitioned into A, B. Let $K \subseteq A$ be s.t. $\mathcal{V} \times (E - K)$ has B as a separator. If $\mathbf{x}$ on S is s.t. $\mathbf{x}/A \in \mathcal{V} \ . \ A$, $x/B \cup K \in \mathcal{V} \ . \ (B \cup K)$, then $\mathbf{x} \in \mathcal{V}$.*

Remark: i. We have described a 'sufficient' procedure for dualization. If the procedure fails we cannot be sure that the 'dual' statement is necessarily false. The procedure is, however, applicable wherever duality is found - we merely have to use the appropriate dual objects and operations.

ii. If we restrict ourselves to the class of planar graphs we have the interesting result that there exists a graph $\mathcal{G}^*$ s.t. $\mathcal{V}_i(\mathcal{G}) = \mathcal{V}_v(\mathcal{G}^*)$. In this case a wider range of statements can be dualized. In particular there is the notion of 'mesh' or 'window' that is dual to that of a vertex. In this book we do not exploit 'planar duality'.

2.8 Notes

Graph theory means different things to different authors. The kind of graph theory that electrical networks need was developed systematically (for quite different reasons) by Whitney [Whitney32], [Whitney33a], [Whitney33b], [Whitney33c]. In this chapter the emphasis has been on the topics of direct use to us later on in the book. We have followed, in the main, [Seshu+Reed61] and [Tutte65] for the sections dealing with graphs, their representations and operations on graphs and vector spaces. For the section on graph algorithms we have used [Aho+Hopcroft+Ullman74] and [Kozen92]. For a recent survey of graph algorithms, the reader is referred to
[Van Leeuwen90].

2.9 Solutions of Exercises

E 2.1: If the graph is disconnected concentrate on one component of it. For this component there are $(n-1)$ possible values for the degree of a node and n vertices if $n > 1$.

E 2.2:
i. If we add all the degrees (counting self loops twice) each edge is being counted twice.

ii. The sum of all degrees is even. So is the sum of all even degrees. So the sum of odd degrees is even and therefore the number of odd degree vertices is even.

E 2.3: (Sketch) Start from any vertex v_o and go to a farthest vertex. If this vertex is deleted there would still be paths from v_o to remaining vertices.

E 2.4:
i. circuit graphs disconnected from each other;
ii. add the edge between a non-terminal vertex and another vertex of the path;
iii. a single edge with two end points;
iv. a graph with only self loop edges.

E 2.5: Consider the graph obtained after deleting the edge e. If v_1, v_2

are two vertices of this graph, there must have been a path P_1 between them in the original graph. If this path had no common vertex with the circuit subgraph it would be present in the new graph also. So let us assume that it has some common vertices with the circuit subgraph. If we go along the path from v_1 to v_2 we will encounter a vertex of the circuit graph for the first time (say the vertex a) and a vertex of the circuit graph for the last time (say b). In the circuit subgraph there is a path P_2 between a and b which does not have e as an edge. If we replace the segment in P_1 between a and b by P_2, we would get a path P_3 in the new graph between v_1 and v_2.

E 2.6: Let P_1, P_2 be the two paths between nodes a, b. We start from node a and go along P_1, P_2 towards b until we reach a vertex say c after which the two paths have different edges. (Note that the vertex c could be a itself). From c we follow P_1 towards b until we reach a vertex say d which belongs also to P_2. Such a vertex must exist since b belongs both to P_1 and to P_2. From d we travel back towards a along P_2. The segments c to d along P_1 and d to c along P_2 would constitute a circuit subgraph since it would be connected and every vertex would have degree 2.

E 2.7: If the graph has a self loop then that is the desired circuit. Otherwise we start from any vertex a and travel outward without repeating edges. Since every vertex has degree ≥ 2 if we enter a vertex for the first time we can also leave it by a new edge. Since the graph is finite we must meet some vertex again. We stop as soon as this happens for the first time. Let c be this vertex. The segment (composed of edges and vertices) starting from c and ending back at c is a circuit subgraph.

E 2.8: (a) A graph made up of self loops only.
(b) A single edge with two end points.

E 2.10: A cutset is a set of crossing edges. Hence, it contains a minimal set of edges which when deleted increases the number of components of the graph. Consider any edge set C with the given property. It should be possible to partition the vertices of the graph into two subsets so that all edges between the two subsets are in C since deletion of C increases the number of components of the graph. Thus, we have two collections of subsets each member of which contains a member of the other. Hence, minimal members of both collections must be identical.

E 2.11: All edges are parallel. There may be isolated vertices.

E 2.12:
i. Deletion of T must increase the number of components. Minimality implies that only one of the components should be split.

ii. if the subgraphs on V_1, V_2 are not connected deletion of T would increase the number of components by more than one. On the other hand, if subgraphs on V_1, V_2 are connected the set of edges between them must constitute a cutset because their deletion increases the number of components and deletion of a proper subset will leave a connected subgraph with vertex set $V_1 \cup V_2$.

E 2.13:
i. There must be at least one path because of connectedness. More than one path would imply the presence of a circuit by Theorem 2.2.1.

ii. The tree graph cannot have only nodes of degree greater or equal to two as otherwise by Theorem 2.2.2 it will contain a circuit. Hence, it has a node a of degree less than two. Now if it has more than one node, because of connectedness, a has degree one. If we start from a and proceed away from it we must ultimately reach a node b of degree one since the graph is finite and repetition of a node would imply two distinct paths between some two nodes.

E 2.14: Proof of Theorem 2.2.4: The trivial single node graph with no edges is a tree graph. The graph on two nodes with an edge between them is also a tree graph. It is clear that in these cases the statement of the theorem is true. Suppose it is true for all tree graphs on $(n-1)$ nodes. Let t be a tree graph of n nodes. This graph, by Theorem 2.2.2, has a vertex v of degree less than two. If $n > 1$, since t is connected, this vertex has degree 1. If we delete this vertex and the single edge incident on it, it is clear that the remaining graph t' has no circuits. It must also be connected. For, if nodes v_1, v_2 have no path in t', the path between them in t uses v as a nonterminal node which therefore has degree ≥ 2 in t-a contradiction. Thus t' is a tree graph on $(n-1)$ nodes. By induction it has $(n-2)$ edges. So t has $(n-1)$ edges. On the other hand, let $\mathcal{G}$ be a connected graph on n nodes with $(n-1)$ edges. If it contains a circuit, by Lemma 2.2.1 we can delete an edge of the circuit without destroying connectedness of the graph. Repeating this procedure would ultimately give us a graph on n nodes

that is connected but has no circuits. But this would be a tree graph with $(n-1)$ edges. We conclude that $\mathcal{G}$ must itself be a tree graph.
Proof of Corollary 2.2.1: The number of edges $= \sum_{i=1}^{p}(n_i - 1)$, where n_i is the number of nodes of the i^{th} component.

E 2.15: We will only show that maximality implies the subset is a forest (coforest). Suppose the set is maximal with respect to not containing a circuit. Then it must intersect each component of the graph in a tree. For, if not, atleast one more edge can be added without the formation of a circuit. This proves the set is a forest.
Next suppose a set L is maximal with respect to not containing a cutset. Removal of such a set from the graph would leave at least a forest of the graph. However, it cannot leave more edges than a forest for in that case the remaining graph contains a circuit. Let e be in this circuit. Deletion of $L \cup e$ cannot disconnect the graph and so $L \cup e$ contains no cutset – this contradicts the maximality. So removal of L leaves precisely a forest.

E 2.16: Deletion of the edges in a cutset increases the number of components in the graph. Hence, every forest must intersect the cutset (otherwise the corresponding forest subgraph would remain when the cutset is deleted and would ensure that the number of components remains the same).
Removal of edges of a coforest must destroy every circuit as otherwise the corresponding forest would contain a circuit. So a coforest intersects every circuit of the graph.

E 2.17: Proof of Lemma 2.2.2: Let a, b be the end points of the edge e being deleted. Let V_a, V_b be the set of all vertices which can be reached from a, b respectively, by paths in the tree graph which do not use e. Suppose node v is not in V_a or V_b. But the connected component containing v cannot meet V_a or V_b (otherwise v can be reached from a or b by a path) and hence, even if e is put back v cannot be connected to $V_a \cup V_b$ by a path. But this would make the tree graph disconnected. We conclude that $V_a \cup V_b$ is the vertex set of the tree graph. The subgraphs on V_a, V_b are connected and contain no circuits and are therefore tree graphs.

E 2.18: Let a, b be the end points of the edge e being contracted. It is clear that the graph after contraction of an edge e is connected. If it contains a circuit graph this latter must contain the fused node $\{a, b\}$.

But if so there exists a path in the original tree graph between a, b which does not use e. This is a contradiction.

E 2.19: Proof of Theorem 2.2.6: Let f_G denote the subgraph of $\mathcal{G}$ on f. By the definition of a forest subgraph, between the end points of e say n_1, n_2 there must be a path, say P in f_G. Addition of e to f_G creates precisely two paths between n_1, n_2, namely, P and the subgraph on e. The path P has n_1, n_2 of degree 1 and remaining vertices of degree two. Hence addition of e to P will create a connected subgraph in which every vertex has degree two. Now this must be the only circuit subgraph created when e is added to f. For if there are two such subgraphs, e must be a part of both of them since f contains no circuit. Hence they and therefore, f_G must have distinct paths between n_1, n_2 which do not use e. But then by Theorem 2.2.1 there must be a circuit subgraph in f_G - a contradiction.

E 2.20: Proof of Theorem 2.2.7: We will prove the result for a connected graph first. Deletion of an edge of a tree graph must increase its connected components by one by Lemma 2.2.2. Deletion of $e \cup \bar{f}$ from the given graph $\mathcal{G}$ is equivalent to first deleting $\bar{f}$ and then, in the resulting tree subgraph f_G on f, deleting e. Therefore, the number of connected components must increase precisely by one when $e \cup \bar{f}$ is deleted. Let a, b be the endpoints of e and let V_a, V_b be the vertex sets of the tree subgraphs (which do not however correspond to trees of $\mathcal{G}$) that result when the edge e is deleted from f_G, equivalently, when $e \cup \bar{f}$ is deleted from $\mathcal{G}$. Any crossing edge set that $e \cup \bar{f}$ contains must have V_a, V_b as end vertex sets. There is only one such. We conclude that $e \cup \bar{f}$ contains only one crossing edge set. This must be a cutset since the subgraphs on V_a, V_b are connected.
If the graph were disconnected, when $e \cup \bar{f}$ is deleted, only one component say $\mathcal{G}_e$ which contains e would be split. Since any cutset contained in $e \cup \bar{f}$ is contained in $\mathcal{G}_e$ we could argue with $\mathcal{G}_e$ in place of $\mathcal{G}$ and the subset of f in $\mathcal{G}_e$ in place of f. So the theorem would be true in this case also.

E 2.21: Let C be a circuit. Let $e \in C$. Then $C - e$ does not contain a circuit and can be grown to a forest of $\mathcal{G}$. C is an f-circuit of this forest.

E 2.22: Let B be a cutset with $e \in B$. By minimality, deletion of $B - e$ will not increase the number of components of the graph, i.e., there is a forest remaining when $B - e$ is deleted. So $B - e$ can be included in a coforest and B is an f-cutset of the corresponding forest.

E 2.23: Let $\hat{f}$ be a forest subgraph of the given graph containing edge e of cutset C.This is possible since e is not a self loop. Contraction of this edge would convert $\hat{f}$ to a forest subgraph f of the new graph. The number of edges in the coforest would not have changed.

E 2.25: Given such a matrix associate a vertex with each row and an edge with each column. The edge has an arrow leaving the vertex (row) where its column has a $+1$ and entering the vertex (row) where its column has a -1. If the column has only zeros the corresponding edge is a self loop incident on any of the vertices.

E 2.31: The matrix retains the property given in Exercise 2.25 when these operations are performed.

E 2.35:
i. When the vertex v is not a cutvertex (i.e., a vertex which lies in every path between some two vertices a, b of the graph which are distinct from itself). In this case deletion of the edges incident at the vertex would break up the graph into atleast three components viz. v alone, component containing a and component containing b.

ii. Consider a graph made up of only two parallel edges.

iii. No. It then has to be orthogonal to itself. Over the real field this would imply that it has null support.

E 2.36: Scan the columns from left. Pick the first column corresponding to a non-selfloop edge. If k columns (edges) have been picked, pick the next column to be corresponding to the first edge which does not form a circuit with previously picked edges. Continue until all columns are exhausted. This gives us a forest of the graph. The f-cutset matrix of this forest with columns in the same order as before and rows such that an identity matrix appears corresponding to the forest would constitute the first set of rows of the RRE matrix. The second set of rows would be zero rows equal in number to the number of components.

E 2.37: If the graph is connected all nodes must have the same potential in order that the voltages of all branches are zero. (Otherwise we can collect all nodes of a particular voltage inside a surface. At least one branch has only one endpoint within this surface. This branch would be assigned a nonzero voltage by the voltage vector). If the graph is disconnected all nodes of the same component must have the same potential by the above argument.

E 2.38: We use the above solution. If the graph is connected we see that $\lambda^T \mathbf{A} = \mathbf{0}$ iff λ has all entries the same, i.e., iff λ belongs to the one dimensional vector space spanned by $(1\ 1 \cdots 1)$. But this means $(\mathcal{C}(A))^{\perp}$ has dimension one. Hence $\dim(\mathcal{C}(A)) = n - 1$, i.e., $r(A) = n - 1$.

E 2.40: Let $\mathbf{i}$ be a nonzero current vector. Let T be the support of $\mathbf{i}$. The subgraph $\mathcal{G} \ . \ T$ of $\mathcal{G}$ must have each vertex of degree at least two (otherwise the corresponding row of $\mathbf{A}$ cannot be orthogonal to $\mathbf{i}$). Hence $\mathcal{G} \ . \ T$ contains a circuit by Theorem 2.2.2. Thus support of $\mathbf{i}$ contains a circuit. Next every circuit vector is a current vector (Theorem 2.3.1). It follows that its support cannot properly contain the support of another nonzero current vector since a circuit cannot properly contain another circuit.
Next let $\mathbf{i}$ be an elementary current vector. Clearly its support must be a circuit C. Let $\mathbf{i}_C$ be the corresponding circuit vector. Now by selecting a suitable scalar α, the current vector $\mathbf{i} + \alpha \mathbf{i}_C$ can be made to have a support properly contained in C. But this implies that the support of $\mathbf{i} + \alpha \mathbf{i}_C$ is void, i.e., $\mathbf{i} = -\alpha \mathbf{i}_C$ as needed.
Now regarding the cutset vector. Let $\mathbf{v}$ be a voltage vector. We know that it must be derived from a potential vector. Let V_1 be the set of all nodes having some fixed potential (among the values taken by the potential vector). Then the crossing edge set corresponding to $(V_1, E - V_1)$ must be a subset of the support of $\mathbf{v}$. Thus, the support of $\mathbf{v}$ must contain a cutset. Now every cutset vector is a voltage vector (Theorem 2.3.2). It follows that its support cannot properly contain the support of another nonzero voltage vector since a cutset cannot properly contain another cutset.
Next let $\mathbf{v}$ be an elementary voltage vector. Proceeding analogously to the current vector case we can show that $\mathbf{v}$ must be a scalar multiple of a cutset vector, as required.

E 2.41: A set of columns T of $\mathbf{A}$ are linearly dependent iff there exists a vector $\mathbf{i}$ with support T such that $\mathbf{Ai} = \mathbf{0}$. By definition $\mathbf{i}$ is a current vector. By Theorem 2.3.7 we know that T must contain a circuit of $\mathcal{G}$. Further, if T contains a circuit of $\mathcal{G}$ the corresponding circuit vector of

$\mathcal{G}$ is a current vector from which it follows that the set of columns T of $\mathbf{A}$ are linearly dependent.
The rows of $\mathbf{B}_f$ constitute a basis for $\mathcal{V}_i(G)$. By the strong form of Tellegen's Theorem we know that $\mathcal{V}_v(G) = (\mathcal{V}_i(G))^{\perp}$. Hence, $\mathbf{v}$ is a voltage vector iff $\mathbf{B}_f\mathbf{v} = \mathbf{0}$. The rest of the argument parallels that of the linear dependence of columns of $\mathbf{A}$.

E 2.42: An f-cutset matrix $\mathbf{Q}_f$ of $\mathcal{G}$ is a representative matrix of $\mathcal{V}_v(G)$ since by Theorem 2.3.2 its rows are linearly dependent on the rows of the incidence matrix and its rank equals the rank of $\mathbf{A}$. Now we know that (Theorem 2.3.8) the columns of $\mathbf{A}$ are linearly independent iff the corresponding edges do not contain a circuit. This must also be true of any representative matrix $\mathbf{Q}_f$ of $\mathcal{V}_v(\mathcal{G})$ since $\mathbf{A}$ and $\mathbf{Q}_f$ have the same column dependence structure. Let $\mathbf{Q}'$ be any standard representative matrix of $\mathcal{V}_v(G)$. Let us assume without loss of generality that

$$\mathbf{Q}' = \begin{array}{cc} T & E-T \\ \big(\ \mathbf{I} & \mathbf{Q}'_{12} \ \big) \end{array}. \qquad (2.10)$$

The columns corresponding to T are linearly independent and $(n-p)$ in number. Hence, T must be a forest of $\mathcal{G}$. Let $\mathbf{Q}_T$ be the f-cutset matrix with respect to T. Then $\mathbf{Q}_T = \begin{pmatrix} \mathbf{I} & \mathbf{Q}_{12} \end{pmatrix}$ for some $\mathbf{Q}_{12}$. But $\mathbf{Q}'$ and $\mathbf{Q}_T$ are row equivalent to each other. So we conclude that $\mathbf{Q}_{12} = \mathbf{Q}'_{12}$ and $\mathbf{Q}' = \mathbf{Q}_T$. The f-circuit case proof is similar.

E 2.44: Proof of Theorem 2.3.9: Each KVE has the form $\mathbf{c}^T\mathbf{v} = \mathbf{0}$, where $\mathbf{c}$ is a circuit vector.
Now every circuit vector is a current vector. So the size of a maximal independent set of circuit vectors cannot exceed $r(\mathcal{V}_i(G))$. However, the rows of $\mathbf{B}_f$ constitute an independent set of circuit vectors of this size. The result follows.

E 2.47:
i. is immediate.

ii. (Sketch) If we start from any node of a circuit subgraph of $\mathcal{G}$ (that intersects T) and go around it, this would also describe an alternating sequence (without edge repetition) of $\mathcal{G} \times T$ starting and ending at the same vertex. This subgraph of $\mathcal{G} \times T$ has each vertex of degree ≥ 2 and so contains a circuit of $\mathcal{G} \times T$. On the other hand given any circuit subgraph of $\mathcal{G} \times T$ we can trace a closed alternating sequence around it which can be expanded to a closed alternating sequence of $\mathcal{G}$ corresponding to a circuit subgraph. So every circuit of $\mathcal{G} \times T$ is the intersection of some circuit of $\mathcal{G}$ with T.

E 2.48: (Sketch) Assume without loss of generality that $\mathcal{G}$ is connected. Any cutset of $\mathcal{G}$ that intersects T would, when removed, increase the number of components of $\mathcal{G} \,.\, T$. Hence, it contains a cutset B_T of $\mathcal{G} \,.\, T$. Any cutset of $\mathcal{G} \,.\, T$ corresponds to vertex sets V_1, V_2 between which it lies (the subgraphs of $\mathcal{G} \,.\, T$ on V_1, V_2 are connected). Now let V_1 be grown to as large a vertex subset of V_1' of $(V(\mathcal{G}) - V_2)$ as possible using paths that do not intersect B_T. Next let V_2 be grown to as large a vertex subset of V_2' of $(V(\mathcal{G}) - V_1')$ as possible using paths that do not intersect B_T. The cutset of $\mathcal{G}$ defined by $V_2', (V(\mathcal{G}) - V_2')$ intersects T in B_T.
Next consider any cutset C_T of $\mathcal{G} \times T$. This corresponds to a partition V_{1T}, V_{2T} of $V(\mathcal{G} \times T)$. Now V_{1T}, V_{2T} are composed of supernodes of $\mathcal{G}$ which are the vertex sets of components of $(\mathcal{G} open T)$. The union of these supernodes yields a partition V_1, V_2 of $V(\mathcal{G})$. Clearly C_T is the set of edges between V_1, V_2. The subgraphs of $\mathcal{G} \times T$ on V_{1T}, V_{2T} are connected. So the subgraphs of $\mathcal{G}$ on V_1, V_2 are also connected. So C_T is a cutset of $\mathcal{G}$. Any cutset of $\mathcal{G}$ made up only of edges in T can similarly be shown to be a cutset of $\mathcal{G} \times T$.

E 2.50: $\mathbf{Ai} = \mathbf{J}$ has a solution iff $\lambda^T \mathbf{A} = \mathbf{0} \Rightarrow \lambda^T \mathbf{J} = \mathbf{0}$. If the graph is connected $\lambda^T \mathbf{A} = \mathbf{0} \Rightarrow$ all components of λ are identical.

E 2.51:
i. A vector satisfies KC Equations of $\mathcal{G} \,.\, T$ iff when padded with $0s$ corresponding to edges in $E(\mathcal{G}) - T$ it satisfies the KC Equations of $\mathcal{G}$. Hence, $\mathcal{V}_i(\mathcal{G} \,.\, T) = (\mathcal{V}_i(\mathcal{G})) \times T$.

ii. Let $\mathbf{i}_T \in \mathcal{V}_i(\mathcal{G} \times T)$. In the graph $\mathcal{G}$ this vector satisfies generalized KCE at supernodes which are vertex sets of components of $\mathcal{G} open T$. The previous exercise implies that we can extend this vector to edges within each of these components. Thus there is a vector $\mathbf{i} \in \mathcal{V}_i(\mathcal{G})$ s.t. $\mathbf{i}/T \in \mathcal{V}_i(\mathcal{G} \times T)$. Thus, $\mathcal{V}_i(\mathcal{G} \times T) \subseteq (\mathcal{V}_i(\mathcal{G})) \cdot T$. Any vector that satisfies KCE of $\mathcal{G}$ would satisfy generalized KCE at supernodes. Hence, if $\mathbf{i} \in \mathcal{V}_i(\mathcal{G})$ then $\mathbf{i}/T \in \mathcal{V}_i(\mathcal{G} \times T)$. Hence, $(\mathcal{V}_i(\mathcal{G})) \cdot T \subseteq \mathcal{V}_i(\mathcal{G} \times T)$.

E 2.52:

i. From Theorem 2.4.6, $\mathbf{R}_{33}$ is a representative matrix of $\mathcal{V} \times T_3$ and

$$\begin{array}{cc} T_1 & T_2 \\ \end{array}$$
$$\left[\begin{array}{cc} \mathbf{R}_{11} & \mathbf{R}_{12} \\ \mathbf{R}_{21} & \mathbf{0} \end{array}\right] \qquad (2.11)$$

is a representative matrix of $\mathcal{V} \cdot (T_1 \cup T_2)$.
Now $\mathbf{R}_{21}, \mathbf{R}_{12}$ are given to have linearly independent rows. So $\mathbf{R}_{21}, \mathbf{R}_{12}$ are representive matrices of $\mathcal{V} \cdot (T_1 \cup T_2) \times T_1$ and $\mathcal{V} \cdot (T_1 \cup T_2) \cdot T_2 (= \mathcal{V} \cdot T_2)$ respectively. Next

$$\begin{array}{cc} T_1 & T_3 \\ \end{array}$$
$$\left[\begin{array}{cc} \mathbf{R}_{21} & \mathbf{R}_{23} \\ \mathbf{0} & \mathbf{R}_{33} \end{array}\right] \qquad (2.12)$$

must be a representive matrix of $\mathcal{V} \times (T_1 \cup T_3)$. So $\mathbf{R}_{21}$ is a representative matrix of $\mathcal{V} \times (T_1 \cup T_3) \cdot T_1$.

ii. If $\mathbf{R}_{11}$ is a zero matrix, then $\mathcal{V} \cdot (T_1 \cup T_2)$ would have T_1, T_2 as separators.

E 2.53: $\mathbf{R}_{33}$ is a representative matrix of $\mathcal{V} \times T_2$ while $\left[\begin{array}{c} \mathbf{R}_{22} \\ \mathbf{R}_{33} \end{array}\right]$ is a representative matrix of $\mathcal{V}\ .\ T_2$. The result follows.

E 2.54:

$$\begin{aligned} \xi'(T) &= r(\mathcal{V}^{\perp} \cdot T) - r(\mathcal{V}^{\perp} \times T) \\ &= \mid T \mid - r(\mathcal{V} \times T) - \mid T \mid + r(\mathcal{V} \cdot T) \end{aligned}$$

(by Theorem 2.4.3).

E 2.55: We shall show that the union of a forest f_1 of $\mathcal{G}short(E-T)$ and a forest f_2 of $\mathcal{G}openT$ yields a forest of $\mathcal{G}$. $\mathcal{G}openT$ has a number of connected components. The forest f_2 intersects each of these components in a tree. The vertex sets (supernodes) V_i of these components $\mathcal{G}_i$ figure as nodes of $\mathcal{G}short(E-T)$. If $f_1 \cup f_2$ contains a circuit of $\mathcal{G}$ it cannot be contained entirely in $\mathcal{G}openT$. The corresponding circuit subgraph can be traced as a closed path starting from some vertex in V_i going through other sets V_j and returning to V_i. When the V_i are fused to single nodes this subgraph would still contain two distinct paths between any pair of its vertices (which are supernodes in the old graph $\mathcal{G}$). Thus, f_1 would contain a circuit of $\mathcal{G}short(E-T)$ which is a contradiction. Hence, $f_1 \cup f_2$ contains no circuit of $\mathcal{G}$. On the other hand we can travel from any vertex v_1 in $\mathcal{G}$ to any other vertex v_f in the same component using only edges of $f_1 \cup f_2$. This is because a connected component of $\mathcal{G}$ would reduce to a connected component of $\mathcal{G}short(E-T)$. So v_1, v_f would be present in supernodes say V_1, V_f which are nodes of $\mathcal{G}short(E-T)$ and which have a path between them using only the edges of f_1. This path P_2 can be exploded into a path P_{12} using only edges of $f_1 \cup f_2$ in $\mathcal{G}$ as follows:

The path P_2 can be thought of as a sequence

$$v_1, v_{11}, e_1, v_2, v_{22}, e_2 \cdots e_f, v'_f, v_f$$

where v_1, v_{11} belong to the same component and in general v_j, v_{jj} belong to the same component of $\mathcal{G}openT$. So would (for notational convenience) v'_f, v_f. Now we can travel from v_1 to v_{11}, v_2 to v_{22}, v_j to v_{jj} etc. using edges of f_2. Addition of these intermediate edges and vertices yields the path P_{12}. Thus, $f_1 \cup f_2$ contains no circuits and contains a tree of each component of $\mathcal{G}$.

E 2.56: Immediate from the above.

E 2.57: Consider the incidence matrix $\mathbf{A}$ of $\mathcal{G}$. A set of columns of $\mathbf{A}$ are linearly independent iff the corresponding edges do not contain a circuit. Thus, T is a separator of $\mathcal{G}$ iff there is no minimal dependent set of columns of $\mathbf{A}$ intersecting both T and $(E-T)$.
Let

$$\begin{array}{cc} & \begin{array}{cc} T & E-T \end{array} \\ \mathbf{R} = & \left[\begin{array}{cc} \mathbf{R}_{TT} & \mathbf{R}_{T2} \\ \mathbf{0} & \mathbf{R}_{22} \end{array}\right] \end{array} \qquad (2.13)$$

be a representative matrix of $\mathcal{V}_v(\mathcal{G})$. This matrix and the incidence matrix are row equivalent and therefore have the same column dependence structure. If the rows of $\mathbf{R_{T2}}$ are linearly dependent on the rows of $\mathbf{R_{22}}$, we can perform reversible row operations using the rows of the latter so that rows of $\mathbf{R_{T2}}$ are made zero.If $\mathbf{R_{T2}}$ is the zero matrix it is clear that no minimal dependent set of columns can intersect both T and $E-T$,where $E \equiv E(\mathcal{G})$. If rows of $\mathbf{R_{T2}}$ are not linearly dependent on those of $\mathbf{R_{22}}$, then $r((\mathcal{V}_v(G)) \cdot T) > r((\mathcal{V}_v(G)) \times T)$. Now, let f_1, f_2 be forests of $\mathcal{G} \,.\, T, \mathcal{G} \,.\, (E-T)$, respectively. The union of these two forests contains more edges than the rank of $\mathcal{G}$ and therefore, contains a circuit. But f_1, f_2 do not individually contain circuits. We conclude that there must exist a circuit that intersects both f_1 and f_2. Thus, we see that T is a separator of $\mathcal{G}$ iff rows of $\mathbf{R_{T2}}$ are linearly dependent on the rows of $\mathbf{R_{22}}$ i.e., iff T is a separator of $\mathcal{V}_v(G)$, i.e., iff $r((\mathcal{V}_v(G)) \cdot T) = r((\mathcal{V}_v(G)) \times T)$. The last statement is equivalent to saying $r(\mathcal{G} \,.\, T) = r(\mathcal{G} \times T)$.

E 2.58: The graph $\mathcal{G}$ has $\alpha_1\alpha_2$ forests as well as coforests, $\beta_1 + \beta_2$ circuits, $\gamma_1 + \gamma_2$ cutsets. This is because every forest of $\mathcal{G}$, when T is a separator, is a union of a forest of $\mathcal{G} \,.\, T$ and a forest of $\mathcal{G} \,.\, (E-T)$. Further, each circuit of $\mathcal{G}$ is either a circuit of $\mathcal{G} \,.\, T$ or a circuit of $\mathcal{G} \,.\, (E-T)$.

E 2.59: Let the directed crossing edge set have the orientation (V_1, V_2). The tail of the edge e lies in a component of the subgraph on V_1.Let V' be the vertex set of this component.Consider the directed crossing edge set defined by $(V', V(\mathcal{G}) - V')$. The head of the edge e lies in a component of the subgraph on $V(\mathcal{G}) - V'$.Let V" be the vertex set of this component.Consider the crossing edge set defined by $(V(\mathcal{G}) - V", V")$.This has e as a member. It can be seen that it is a directed cutset.

E 2.60: We use Kuhn-Fourier Theorem. Let $\mathcal{V}$ be the solution space of $\mathbf{Ax} = \mathbf{0}$. Suppose $\mathcal{V}$ has no nonnegative vector whose support contains e. Then the following system of inequalities has no solution

$$\mathbf{Ax} = \mathbf{0}$$

$$\mathbf{x}(e) > 0$$

$$\mathbf{x} \geq \mathbf{0}.$$

By Kuhn-Fourier Theorem there exists a vector λ, a scalar $\alpha > 0$ and a vector $\sigma \geq \mathbf{0}$ s.t. $\lambda^{\mathbf{T}}\mathbf{A} + \alpha\chi_{\mathbf{e}} + \sigma^{\mathbf{T}} = \mathbf{0}$. Thus, $-\lambda^{\mathbf{T}}\mathbf{A} = (\sigma^{\mathbf{T}} + \alpha\chi_{\mathbf{e}})$. The vector $\sigma^T + \alpha\chi_e$ lies in the space $\mathcal{V}^{\perp}$ and has e in its support.

E 2.61:
ii. From each vertex obtain the set of all reachable vertices (do a bfs). This takes $O(\mid V \mid\mid E \mid)$ time. Sort each of these sets and obtain a list in increasing order of indices. This takes $O(\mid V \mid^2 \log \mid V \mid)$ time. For each pair (v_1, v_2) check if v_2 is reachable from v_1 and if v_1 is

reachable from v_2. This takes $O(| V |^2)$ time. So overall complexity is $O(| V | (\max(| E |, | V | \log | V |)))$.

E 2.62: We assume that the length of an edge is an integer. We first find an upper bound u and a lower bound l for the length of this path. The upper bound could be the sum of all the lengths and the lower bound could be the minimum length of an edge. Narrow down to the correct value of the distance between v_1 and v_2 by asking question of the type 'is there a path between v_1 and v_2 of length $\leq d_i$'. The value of d_i in this question could be chosen by binary search between u and l: $d_1 = (l + \frac{u-l}{2})$, if yes $d_2 = (l + \frac{u-l}{4})$, if no $d_2 = (u - \frac{u-l}{4})$ and so on. (Whenever any of these numbers is a fraction we take the nearest integer). Clearly the number of such d_i is $O(\log(u - l))$.
Suppose d is the length of the shortest path. To find the edges of the shortest path we ask, for each edge e between v_1 and v_{11} say, if there is a path of length $d - d(e)$ between v_{11} and v_2. If yes (and it must be yes for one such edge) then e belongs to the path and we now try to find a path of length $d - d(e)$ between v_{11} and v_2. By this process the shortest path can be found by framing $O(| E(\mathcal{G}) |)$ decision problems. Overall the total number of decision problems is $O(\log(u - l) + | E(\mathcal{G}) |)$.

E 2.63: Observe that in the stack at any stage the top vertex has the highest dfs numbering and we cannot get below it unless it has been deleted from the stack. Once a vertex has been deleted from the stack it can never reappear. If v_1 is not an ancestor of v_2 then they have a common ancestor v_3 of highest dfs number. Since v_1 has a lower dfs number than v_2 it would have been deleted from the stack before we went back to v_3 and travelled down to v_2. But then the edge e would have been scanned when we were processing v_1 for the last time. At that time the other end of e would have been unmarked and e would then have been included in the dfs tree. This is a contradiction.

E 2.64: The technique described for building f-circuits using dfs would work for any rooted tree (a tree in which each node has a single parent). In the case of bfs we walk from v_1 and v_2 towards the root by first equalising levels (if v_1 has a higher level number we first reach an ancestor v_1' of the same level as v_2). Thereafter we move alternately one step at a time in the paths v_1 to root and v_2 to root until the first common ancestor is reached.

E 2.65: Let the sequence of edges generated by Prim's algorithm in building t_{alg} be $e_1, e_2, \ldots, e_k$. Let t be a min spanning tree which has the longest unbroken first segment $e_1, e_2, \ldots, e_r$ in common with t_{alg}. We will show that $r = k$. Suppose $r < k$. Now e_{r+1} was selected during the execution of the algorithm as the edge of least weight with one end in the current set of vertices $V(\{e_1, e_2, \ldots, e_r\})$ and another in the complement. Consider the f-circuit $L(e_{r+1}, t)$. The edges of $L(e_{r+1}, t) - e_{r+1}$ constitute a path between the endpoints of e_{r+1}. Atleast one of them, say $\hat{e}$, has only one endpoint in $V(\{e_1, e_2, \ldots, e_r\})$ and has weight not less than that of e_{r+1}. Now $t - \hat{e} \cup e_{r+1}$ is a tree with weight greater than that of t and a greater first segment overlap with t_{alg}. This is a contradiction.

Suppose t is a minimum spanning tree whose total weight is less than that of the tree t_{alg} generated by the algorithm. Let t be the nearest such tree to t_{alg} (*i.e.*, $\mid t_{alg} - t \mid$ is minimum). Let $e \in (t_{alg} - t)$. Consider the f-circuit $L(e, t)$. If $w(e) \leq w(e_j)$ for some $e_j \in (L(e, t) - e)$, then we could replace t by the tree $t \cup e - e_j$ without increasing its weight. This would contradict the fact that t is the nearest minimum spanning tree to t_{alg}. Hence $w(e) > w(e_j)$ for each e_j in $(L(e, t) - e)$. However, e was selected, during some stage of the algorithm, as the edge of least weight with one end in the current set of vertices V_e. The edges of $(L(e, t) - e)$ constitute a path between the end points of e. At least one of them, therefore, has only one end point in V_e and, therefore, has weight not less than that of e. This contradiction proves that $t_{alg} - t$ is void. Since both t_{alg} and t have the same number of edges we conclude that $t_{alg} = t$.

E 2.67: Construct a 'level graph' containing all the edges of a bfs tree in the residual graph from s to t and any other edge of that graph that travels from a lower to a higher level. (The level of a node is the bfs number of the node). Clearly only such edges can figure in a shortest path from s to t. Whenever we augment the flow using a shortest path upto its bottleneck capacity, atleast one of the edges, say e, of the residual graph will drop out of the level graph.
In the residual graph an oppositely directed edge to e would remain. But this edge cannot figure in the level graph unless the length of the shortest path changes (increases), since it would be travelling from a higher to a lower level. An edge that has dropped out cannot return until the length of the shortest path changes. It follows that there

can be at most m augmentations at a particular length of the shortest path from s to t. The length of the shortest path cannot decrease and also cannot exceed the number of nodes in the graph. Hence the total number of augmentations cannot exceed mn.

E 2.68: (Sketch) Replace each edge by two oppositely directed edges of capacity 1. Treat s as source and t as sink. Maximize flow from source to sink. Each unit of flow travels along a path whose edges (since their capacity is 1) cannot be used by another unit of flow. Hence, the maximum flow $\leq$ maximum number of arc disjoint paths. The reverse inequality is obvious. In any cut of the flow graph the forward arcs (each of capacity 1) would correspond to arcs in the corresponding cut of the original graph. The result follows.

E 2.69:
i. $r(T)$ = size of the maximum circuit free set contained in T. So the dual function at T would give the size of the maximum cutset free set contained in T, i.e., the dual is $\nu(\cdot)$, the nullity function ($\nu(T) =\mid T \mid$ $-r(\mathcal{G} \times T)$).
ii. Let $r^*(\cdot)$ be the dual. Then

$$r^*(T) \equiv \dim(\mathcal{V}^{\perp} \cdot T) =\mid T \mid - \dim(\mathcal{V} \times T)$$

iii. Let $\xi^*(\cdot)$ be the dual. Then

$$\begin{aligned}\xi^*(T) \equiv \dim(\mathcal{V}^{\perp} \cdot T) - dim(\mathcal{V}^{\perp} \times T) &= \mid T \mid -dim(\mathcal{V} \times T) - \mid T \mid +dim(\mathcal{V} \,.\, T) \\ &= \xi(T).\end{aligned}$$

Thus, $\xi(\cdot)$ is self dual.
iv. Closed sets are complements of unions of cutsets. So the duals are complements of unions of circuits.
v. Selfloop is a single edged circuit. The dual is a single edged cutset, i.e., a coloop.
vi. the dual is the selfloop.
vii. A separator T satisfies $r(\mathcal{V} \,.\, T) - r(\mathcal{V} \times T) = 0$, i.e., $\xi(T) = 0$. Its dual would satisfy $\xi^*(T) = 0$. But we saw that $\xi(T) = \xi^*(T)$. So the dual of 'separator' is 'separator'.
viii. A separator T satisfies $r(\mathcal{G} \,.\, T) - r(\mathcal{G} \times T) = 0$, i.e.,

$$r((\mathcal{V}_v(\mathcal{G})) \cdot T) - r((\mathcal{V}_v(\mathcal{G})) \times T = 0.$$

If we go through the procedure of dualization we must replace $\mathcal{V}_v(\mathcal{G})$ by $\mathcal{V}, \mathcal{V}$ by $\mathcal{V}^\perp$. This would yield

$$r(\mathcal{V}^\perp \cdot T) - r(\mathcal{V}^\perp \times T) = 0.$$

As we have seen before this is equivalent to

$$r(\mathcal{V} \,.\, T) - r(\mathcal{V} \times T) = 0.$$

Substituting $\mathcal{V}_v(\mathcal{G})$ in place of $\mathcal{V}$ and interpreting in terms of $\mathcal{G}$ we get

$$r(\mathcal{G} \,.\, T) - r(\mathcal{G} \times T) = 0$$

Thus separator of a graph is self dual.

E 2.70:

i. replace 'coforest' by 'forest', 'circuit' by 'cutset'.
ii. as above.
iii. replace 'circuits' by 'cutsets'.
iv. replace 'circuit' by 'cutset'.
v. replace 'forest' by 'coforest', interchange 'dot' and 'cross' operations.
vi. replace '$r(\cdot)$' by '$\nu(\cdot)$', interchange 'dot' and 'cross' operations, replace 'self loops' by 'coloops'
vii. interchange 'dot' and'cross', $\mathcal{V}$ and $\mathcal{V}^\perp$.

The dual is true if the original is true (and the original is in fact true) in each of the above cases.

2.10 Solutions of Problems

P 2.1: (Sketch) Break the graph up into disjoint union of circuit subgraphs. This is possible since when a circuit is deleted the remaining graph still has only even degree vertices. Within each circuit subgraph we can start from any vertex, go through all vertices and come back to it. By induction, when one circuit is deleted, in the remaining graph within each component we can start from any vertex go through all vertices and come back to it. Now start from a vertex of the (deleted) circuit subgraph, go around it until a vertex of one of the components of the remaining graph is met. Complete a closed traversal of the component, come back to the vertex of the circuit subgraph and proceed along the circuit subgraph until the next vertex of a component is met. Continue until you come back to the starting vertex of the circuit subgraph.

P 2.5:
i. is easy to see.

ii. Start from the two end points of $e_d \in C_1 - C_2$, proceed outward until you first reach vertices a, b of the subgraph on C_2 (a, b could even be the end points of e_c). Vertices a, b must be distinct as otherwise $C_1 \cap C_2 = \emptyset$. Now in the subgraph on C_2 there are precisely two distinct paths between a, b. Only one of them contains e_c. If we follow the other path we would have constructed the desired circuit subgraph corresponding to C_3.

P 2.9: We use the notation of the Circuit Axioms in Problem 2.5. It is easy to see that axiom (i) is satisfied.
Axiom (ii): If (V_1, V_2) defines C_1 and (V_1', V_2') defines C_2, then e_c lies between $V_1 \cap V_1'$ and $V_2 \cap V_2'$ while e_d lies entirely within V_1' or entirely within V_2'. Delete $C_1 \cup C_2$. The graph is broken up into atleast three pieces (atleast three of the sets $V_1 \cap V_1'$, $V_1 \cap V_2'$, $V_2 \cap V_1'$, $V_2 \cap V_2'$ must be nonvoid). Now add back e_c . The graph would still have atleast two pieces. Consider the crossing edge set corresponding to $((V_1 \cap V_1') \cup (V_2 \cap V_2'), (V_1 \cap V_2') \cup (V_2 \cap V_1'))$. This crossing edge set contains e_d and is itself contained in $C_1 \cup C_2$. Now every crossing edge set is a disjoint union of cutsets (Problem 2.11). So there exists a cutset C_3 s.t. $e_d \in C_3 \subseteq C_1 \cup C_2 - e_c$.

P 2.11: Let $\mathcal{G}$ be connected. Let (V_1, V_2) define the crossing edge set. Let the subgraph on V_1 have components whose vertex sets are $V_{11}, \cdots, V_{1k}$ and V_2 be similarly partitioned into $V_{21}, \cdots, V_{2t}$. When $k = t = 1$ the result is clear since the crossing edge set is a cutset.Otherwise we can break up the crossing edge set into crossing edge sets corresponding to $(V_{11}, (V_2 \cup V_1) - V_{11}), \cdots, (V_{1k}, (V_2 \cup V_1) - V_{1k})$. So without loss of generality we assume $k = 1$. In this case the crossing edge set can be broken up into cutsets corresponding to $(V_{21}, (V_2 \cup V_1) - V_{21}), \cdots, (V_{2t}, (V_2 \cup V_1) - V_{2t})$. (The subgraph on V_{21} is connected and the subgraph on $(V_2 \cup V_1 - V_{21})$ is connected because the graph $\mathcal{G}$ is connected and the subgraph on V_{11} is connected).

P 2.13:
i. By KCE at a node of degree 1, the branch incident at it must carry zero current. So the vertex and the branch can be deleted without affecting KCE at any other node. What is left is a tree graph on a smaller set of nodes so the argument can be repeated.

ii. All selfloops. This is the structure that results when a tree of the (connected) graph is contracted. (Observe that the tree graph results when the cotree is deleted (opened)).

P 2.14: (a) At least one of the non datum nodes has degree 1. This node and the corresponding terminal branch would give us a row and a column which contain only one nonzero entry (± 1, where they meet). Deletion of this node and edge would give us a tree graph on nodes whose size is one less than before. Its determinant could be taken, by induction, to be ± 1. So the original determinant is also ± 1.
(b) For every injecting current vector corresponding to all the non datum nodes, if one can fix the currents in the branches **uniquely** we are done. The current at a tree branch that is terminal (incident at a vertex of degree 1) could be taken as a part of the injecting current source. We are now left with a new tree graph on less number of nodes for which (by induction) we may assume that branch currents are fixed uniquely by injecting currents.
(c) We have $\lambda^T(\mathbf{A}_r) = \mathbf{v}^T$. So if λ^T is uniquely fixed for a given $\mathbf{v}^T$ we are done. Starting from the datum node we travel to a given node along voltage sources (This is possible since the graph is connected). Their algebraic sum gives the **unique** node voltage.

P 2.15: Suppose the support is contained in a forest. Then there is a nontrivial solution to the KCE of a forest graph which is impossible by Problem 2.13 (one can also argue in terms of f-cutsets of this forest).

P 2.16: Suppose the support of the voltage vector meets only the coforest. So we have all the forest voltages zero. But each coforest edge forms an f-circuit with the forest. So its voltage is the algebraic sum of the voltages of the circuit branches in the forest. This would give the coforest edge voltage to be zero. So the voltage vector would be a zero vector.

P 2.17: Use solution of Problem 2.14 (a). Now $A_r = A_{rt}\mathbf{Q}_f$, where $\mathbf{Q}_f$ is the f-cutset matrix corresponding to tree t.From this and from Problem 2.14 conclude that determinant of every full submatrix of $\mathbf{Q}_f$ is $0, \pm 1$. For proving the property for subdeterminants use appropriate trees. (Note that, if a subdeterminant is nonzero, corresponding columns say $t \cap t'$, together with some other edges form a tree t'. The determinant corresponding to t' is ± 1. But this is also equal to ± 1(subdet corresponding to $t \cap t'$) $\times$ (subdet $(t' - t)$). Since both the factors are clearly integers, the result follows.

P 2.18: See solution of Problem 2.14.

P 2.19: (a) The vector $\mathbf{b}$ gives the injected currents at the nodes. Let $\mathbf{Ax} = \mathbf{b}$ be equivalent to $\mathbf{Q}_f\mathbf{x} \equiv \begin{pmatrix} \mathbf{I} & \mathbf{Q}_{12} \end{pmatrix} \begin{matrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{matrix} = \hat{\mathbf{b}}$. One possible solution of the latter equations has all entries in $\mathbf{x}_2$ zero and $\mathbf{x}_1 = \hat{\mathbf{b}}$. This is also a solution of the former equations. So $\hat{\mathbf{b}}$ is the vector of forest branch currents (the forest being the one corresponding to $\mathbf{Q}_f$) when the node injection currents are given by $\mathbf{b}$. To compute $\hat{\mathbf{b}}$ by inspection proceed as follows: (a) select some node as datum node in each component. (b) for each node v draw a path P_v in the forest graph to the datum node of the component. Associate with P_v the value of $\mathbf{b}$ at v. (c) if e is a branch of the forest the value of $\hat{\mathbf{b}}$ at e is the algebraic sum of the $\mathbf{b}(v)$ where e lies in P_v. (If e agrees with P_v add $\mathbf{b}(v)$, if it opposes subtract $\mathbf{b}(v)$ and if e does not lie in P_v ignore $\mathbf{b}(v)$).

ii. Start with the graph $\mathcal{G}\ .\ f_1$. To this add a copy $f_2{}'$ of f_2. Now associate each branch e' of $f_2{}'$ with a current source of value equal to $\mathbf{b}_2(e')$. Find the currents in branches of f_1 by constructing f-circuits of branches in $f_2{}'$ and taking the value of $\mathbf{b}_1(e)$ to be the algebraic sum of $\mathbf{b}_2(e')$ where e lies in the f-circuit of e'.

P 2.20: Consequence of total unimodularity of $\mathbf{Q}_f, \mathbf{B}_f$, i.e., every subdeterminant has value $0, \pm 1$ (see Problem 2.17).

P 2.21: $\mathbf{AA}^T$: The (i, j) entry is the negative of the number of edges

between i, j, if $i \neq j$, and equal to the number of edges incident at i, if $i = j$.
$\mathbf{B}_f\ \mathbf{B}_f{}^T$: The (i, j) entry is (number of edges which lie in i^{th} and j^{th} f-circuits with same orientation relative to f-circuit orientation) - (number of edges which lie in i^{th} and j^{th} f-circuits with opposite orientation).
$\mathbf{Q}_f\ \mathbf{Q}_f{}^T$: similar to $\mathbf{B}_f\ \mathbf{B}_f{}^T$ case.

P 2.22: (sketch) Let $\mathbf{C}_1, \mathbf{C}_2$ be the circuits (cutsets). The corresponding circuit (cutset) vectors can be a part of the same f-circuit (f-cutset) matrix iff $\mathcal{G}\ .\ (\mathbf{C}_1 \cup \mathbf{C}_2)$ $(\mathcal{G} \times (\mathbf{C}_1 \cup \mathbf{C}_2))$ has nullity 2 (rank 2). These ideas follow by noting that if $\mathbf{C}_1, \mathbf{C}_2$ correspond to f-circuit vectors of some forest then the submatrix of the f-circuit matrix of that forest composed of these two vectors and columns $\mathbf{C}_1 \cup \mathbf{C}_2$ must be a representative matrix of $\mathcal{V}_i(\mathcal{G}\ .\ (\mathbf{C}_1 \cup \mathbf{C}_2))$ (using Theorem 2.4.6 and the fact that $\mathcal{V}_i(\mathcal{G}\ .\ T) = (\mathcal{V}_i(\mathcal{G})) \times T$). The cutset case arguments are dual to the above.

P 2.23: See Subsection 2.6.2 for a good algorithm for building the f-circuit.Building all f-circuits of a tree has been shown there to be $O(\sum \mid L(e, t) \mid)$. For building the f-cutset, with respect to a tree t, of a branch e_t, find the sets of all nodes reachable from either of the end points of e_t in the graph $\mathcal{G} \cdot t$ by doing a bfs. If these sets are V_1, V_2 respectively then the desired f-cutset is defined by (V_1, V_2). The complexity of this algorithm is $O(|V(\mathcal{G})|)$. However building all f-cutsets of a tree is clearly equivalent to building all f-circuits of the same tree (see Exercise 2.43).

P 2.24: (Sketch) If $\mathcal{G}'$ is a subgraph of $\mathcal{G}$ then let $\hat{\mathcal{V}}_i(\mathcal{G}')$ denote the vectors obtained from those of $\mathcal{V}_i(\mathcal{G}')$ by adjoining zeros corresponding to edges outside $\mathcal{G}'$. Clearly $\hat{\mathcal{V}}_i(\mathcal{G}_j) \subseteq \mathcal{V}_i(\mathcal{G})$. Now, by construction, there exists a coforest of $\mathcal{G}\ .\ (\bigcup_{t=1}^{j-1} E(\mathcal{G}_t))$ that does not intersect $E(\mathcal{G}_j), j = 2, \cdots, k$. Assume by induction that $\hat{\mathcal{V}}_i(\mathcal{G}\ .\ (\bigcup_{t=1}^{j-1} E(\mathcal{G}_t)))$ has $\mathbf{R}'_{j-1} \equiv \begin{bmatrix} \hat{\mathbf{R}}_1 \\ \vdots \\ \hat{\mathbf{R}}_{j-1} \end{bmatrix}$ as a representative matrix. The rows of $\hat{\mathbf{R}}_j$ are linearly independent of these rows since the columns corresponding to the above coforest are independent in $\mathbf{R}'_{j-1}$ and have zero entries in $\hat{\mathbf{R}}_j$. So if we show that $\mathbf{R}'_j$ has the correct number of rows

$(=\nu(\mathcal{G}\ .\ (\bigcup_{t=1}^{j}E(\mathcal{G}_t))))$ we are done.We have,

$$\nu(\mathcal{G}\ .\ (\bigcup_{t=1}^{j}E(\mathcal{G}_t)))=\nu(\mathcal{G}\ .\ (\bigcup_{t=1}^{j-1}E(\mathcal{G}_t)))+\nu(\mathcal{G}\ .\ (\bigcup_{t=1}^{j}E(\mathcal{G}_t))\times(E(\mathcal{G}_j)-(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))))$$

(using Corollary 2.4.3).
Now

$$\nu(\mathcal{G}\ .\ (\bigcup_{t=1}^{j}E(\mathcal{G}_t))\times(E(\mathcal{G}_j)-(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))))=\nu(\mathcal{G}_j\times(E(\mathcal{G}_j)-(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))))$$

(since,as far as $E(\mathcal{G}_j)$ is concerned, contracting all of $(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))$ is the same as contracting the forest $E(\mathcal{G}_j)\cap\left[\bigcup_{i=1}^{j-1}E(\mathcal{G}_i)\right]$ of $\mathcal{G}\ .\ (\bigcup_{t=1}^{j-1}E(\mathcal{G}_t)))$.
But

$$\nu(\mathcal{G}_j\times(E(\mathcal{G}_j)-(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))))=\nu(\mathcal{G}_j)-\nu(\mathcal{G}_j\cdot(E(\mathcal{G}_j)\cap(\bigcup_{t=1}^{j-1}E(\mathcal{G}_t))))=\nu(\mathcal{G}_j)$$

(since $E(\mathcal{G}_j)\cap\left[\bigcup_{i=1}^{j-1}E(\mathcal{G}_i)\right]$ is a subforest of $\mathcal{G}_j$). This proves the required result.

P 2.25: See [Narayanan85c].

P 2.26: (Sketch)
i. Every forest of $\mathcal{G}$ intersects T in a subforest of $\mathcal{G}\ .\ T$ and every forest of $\mathcal{G}\ .\ T$ can be grown to a forest of $\mathcal{G}$.

ii. Union of a forest of $\mathcal{G}\ .\ (E-T)$ and a forest of $\mathcal{G}\times T$ is a forest of $\mathcal{G}$. The result now follows from the previous part.

iii. 'Only if' is clear. Suppose $K\cup$ (a forest f_{E-T} of $\mathcal{G}\ .\ (E-T)$) is a forest of $\mathcal{G}$. When edges of f_{E-T} are contracted K would not contain a circuit. The remaining edges of $(E-T)$ would have become selfloops by then. So K must also be a subforest of $\mathcal{G}\times T$. But $\mid K\mid=r(\mathcal{G}\times T)$. So K is a forest of $\mathcal{G}\times T$.

iv. Arguments similar to the previous part.

P 2.27: (Sketch)
i. We know that union of a forest of $\mathcal{G}\ .\ A_1$ and a forest of $\mathcal{G}\times(E-A_1)$ is a forest of $\mathcal{G}$ and if a forest of $\mathcal{G}$ contains a forest of $\mathcal{G}\ .\ A_1$ then its intersection with $(E-A_1)$ is a forest of $\mathcal{G}\times(E-A_1)$. So the given statement is true for $n=2$. If it is true for $n=k-1$ then by working

with $\mathcal{G} \, . \, (A_1 \cup \cdots \cup A_{k-1})$ and $\mathcal{G} \times A_k$ we see that it must be true for $n = k$ also.

ii. If the graph has only such forests $A_1, \cdots, A_n$ become separators. Proof by induction.

iii. If this is true for each σ then $A_1, \cdots, A_n$ become separators.

P 2.28: (Sketch for the second part) Select a forest with priority $E - T_1, T_2, T_1 - T_2$. Now use ideas of Exercise 2.52.

P 2.29: We need to check that there is no violation of KCL (KVL) in any cutset (circuit) contained in T. So check if there is violation of KCL in $\mathcal{G} \times T$ and violation of KVL in $\mathcal{G} \, . \, T$.

Bibliography

[Aho+Hopcroft+Ullman74] A.V. Aho, J.E. Hopcroft and J.D. Ullman: *The Design and Analysis of Computer Algorithms* (Addison-Wesley, Reading, Mass., 1974).

[Ahuja+Magnanti+Orlin93] R.K.Ahuja, T.L.Magnanti and J.B.Orlin: *Network Flows: Theory, Algorithms and Applications* (Prentice Hall, Englewood Cliffs 1993).

[Aigner79] M. Aigner: *Combinatorial Theory* (Springer, Berlin, 1979).

[Amari62] S.Amari: Topological foundations of Kron's tearing of electrical networks. RAAG Memoirs **3** (1962) 322.

[Asimow+Roth78] L. Asimow and B. Roth: The rigidity of graphs I. *Transactions of American Mathemaical Society* **245** (1978) 279-289.

[Asimow+Roth79] L. Asimow and B. Roth: The rigidity of graphs II. *Transactions of American Mathemaical Society* **68** (1979) 171-190.

[Balas+Pulleyblank87] E.Balas and W.R.Pulleyblank: The perfectly matchable subgraph polytope of an arbitrary graph: CORR Research Report 87-39, Faculty of Maths, University of Waterloo (1987).

[Belevitch68] V. Belevitch: *Classical Network Theory* (Holden-Day, San Francisco, 1968).

[Berge73] C. Berge: *Graphs and Hypergraphs* (translated by E.Minieka, North Holland, Amsterdam, 1973).

[Birkhoff35] G. Birkhoff: Abstract linear dependence and lattices. *American Journal of Mathematics* **57** (1935) 800-804.

[Birkhoff67] G. Birkhoff: *Lattice Theory* (American Mathematical Colloquium Publications **25** (3rd ed.) Providence, R.I., 1967).

[Bordewijk56] J.L. Bordewijk: Inter-reciprocity applied to electrical networks. *Applied Science Research*, Netherlands **6B** (1956) 1-74.

[Brameller+John+Scott69] A. Brameller,M. John and M. Scott: *Practical Diakoptics for Electrical Networks* (Chapman and Hall, London, England, 1969).

[Branin62] F.H.Branin Jr.: The relationship between Kron's method and the classical methods of network analysis. *Matrix and Tensor Quarterly* **12** (1962) 69-105.

[Brualdi74] R.A. Brualdi: Matroids induced by directed graphs - a survey. In: *Recent Advances in Graph Theory* (Proceedings of the Symposium, Prague, Academia Praha, June 1974) 115-134.

[Bruno+Weinberg71] J. Bruno and L. Weinberg: The principal minors of a matroid. *Linear Algebra and Its Applications* **4** (1971) 17-54.

[Catlin+Grossman+Hobbs+Lai92] P.A.Catlin, J.W.Grossman, A.M.Hobbs and H.J.Lai: Fractional arboricity, strength and principal partitions in graphs and matroids. *Discrete Applied Mathematics* **40** (1992) 285-302.

[Cayley1889] A. Cayley: A theorem on trees. *Quarterly Journal of Pure and Applied Mathematics* **23** (1889) 376-378.

[Choquet55] G. Choquet: Theory of capacities. *Annales de l'Institut Fourier* **5** (1955) 131-295.

[Chua+Lin75] L.O.Chua and P.M.Lin: *Computer - Aided Analysis of Electronic Circuits: Algorithms and Computational Techniques* (Prentice Hall, 1975).

[Chua+Desoer+Kuh87] L.O.Chua, C.A.Desoer and E.S.Kuh: *Linear and Nonlinear Circuits* (McGraw-Hill, New York, 1987).

[Cormen+Leiserson+Rivest90] T.H.Cormen, C.E.Leiserson and R.L.Rivest: *Introduction to Algorithms* (MIT Press, Cambridge, Mass.,1990).

[Crapo+Rota70] H.H. Crapo and G.C. Rota: *On the Foundations of Combinatorial Theory - Combinatorial Geometry* (MIT Press, Cambridge, MA, 1970).

[Cunningham84] W.H. Cunningham: Testing membership in matroid polyhedra. *Journal of Combinatorial Theory* **B36** (1984) 161-188.

[Cunningham85] W.H. Cunningham: On submodular function minimization. *Combinatorica* **5** (1985) 185-192.

[Desoer+Kuh69] C.A.Desoer and E.S.Kuh: *Basic Circuit Theory* (McGraw-Hill, New York, 1969).

[Dilworth44] R.P. Dilworth: Dependence relations in a semimodular lattice. *Duke Mathematical Journal* **11** (1944) 575-587.

[Dinic70] E.A. Dinic: Algorithm for solution of a problem of maximum flow in a network with power estimation. *Soviet Mathematics Doklady* **11** (1970) 1277-1280.

[Director+Rohrer69] S.W. Director and R.A. Rohrer: The Generalized Adjoint Network and Network Sensitivities. *IEEE Transactions on Circuit Theory* **CT-16** (1969) 318-323.

[Dulmage+Mendelsohn59] A.L. Dulmage and N.S. Mendelsohn: A structure theory of bipartite graphs of finite exterior dimension. *Transactions of the Royal Society of Canada*, Third Series, Section III, **53** (1959) 1-13.

[Edmonds65a] J. Edmonds: Minimum partition of a matroid into independent subsets. *Journal of Research of the National Bureau of Standards* **69B** (1965) 67-72.

[Edmonds65b] J. Edmonds: Lehman's switching game and a theorem of Tutte and Nash-Williams. *ibid.* **69B** (1965) 73-77.

[Edmonds68] J. Edmonds: Matroid partition. In: *Mathematics of the Decision Sciences, Part I* (Lectures in Applied Mathematics) **11** (1968) 335-345.

[Edmonds70] J. Edmonds: Submodular functions, matroids, and certain polyhedra. *Proceedings of the Calgary International Conference on Combinatorial Structures and Their Applications* (R.Guy, H.Hanani, N.Sauer and J.Schönheim, eds., Gordon and Breach, New York, 1970), 69-87.

[Edmonds79] J. Edmonds: Matroid intersection. *Annals of Discrete Mathematics.* **4** (1979) 39-49.

[Edmonds+Fulkerson65] J. Edmonds and D.R. Fulkerson: Transversals and matroid partition. *Journal of Research of the National Bureau of Standards***69B** (1965) 147-157.

[Edmonds+Karp72] J.Edmonds and R.M. Karp: Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of ACM* **19** (1972) 248-264.

[Faigle87] U. Faigle: Matroids in combinatorial optimization. In: *Combinatorial Geometries* (N. White, ed., Encyclopedia of Mathematics and Its Applications **29**, Cambridge University Press, 1987) 161-210.

[Ford+Fulkerson56] L.R. Ford and D.R. Fulkerson: Maximal flow through a network. *Canadian Journal of Mathematics* **8** (1956) 399-404.

[Ford+Fulkerson62] L.R. Ford and D.R. Fulkerson: *Flows in Networks* (Princeton University Press, Princeton, N.J., 1962).

[Fourier1826] J. Fourier: Solution d'une question particulière du calcul des inégalités. *Oeuvres II* (1826) 317-328.

[Frank82] A. Frank: An algorithm for submodular functions on graphs. *Annals of Discrete Mathematics* **16** (1982) 189-212.

[Frank+Tardos88] A. Frank and É. Tardos: Generalized polymatroids and submodular flows. *Mathematical Programming* **42** (1988) 489-563.

[Frank94] A.Frank: On the edge connectivity algorithm of Nagamochi and Ibaraki.Laboratoire Artemis,IMAG,Universite J.Fourier, Grenoble,(March 1994).

[Fujishige78a] S. Fujishige: Algorithms for solving the independent-flow problems. *Journal of the Operational Research Society of Japan* **21** (1978) 189-204.

[Fujishige78b] S. Fujishige: Polymatroid dependence structure of a set of random variables. *Information and Control* **39** (1978) 55-72.

[Fujishige80a] S. Fujishige: Lexicographically optimal base of a polymatroid with respect to a weight vector. *Mathematics of Operations Research* **5** (1980) 186-196.

[Fujishige80b] S. Fujishige: Principal structures of submodular systems. *Discrete Applied Mathematics* **2** (1980) 77-79.

[Fujishige84] S. Fujishige: Submodular systems and related topics. *Mathematical Programming Study* **22** (1984) 113-131.

[Fujishige91] S.Fujishige: *Submodular functions and optimization* (Annals of Discrete Maths **47**) (North Holland,Amsterdam,New York,Oxford,Tokyo,1991).

[Gale+Kuhn+Tucker51] D. Gale, H.W. Kuhn and A.W. Tucker: Linear Programming and the theory of games. In: *Activity Analysis of production and allocation* (John Wiley, New York, 1951).

[Gale57] D. Gale: A theorem of flows in networks. *Pacific Journal of Mathematics* **7** (1957) 1073-1082.

[Gale68] D. Gale: Optimal assignment in an ordered set: an application of matroid theory. *J. Combinatorial Theory* **4** (1968) 176-180.

[Gantmacher59] F.R.Gantmacher: *The Theory of Matrices*. Volumes I,II (translated from the Russian) (Chelsea, NewYork, 1959).

[Garey+Johnson79] M.R. Garey and D.S. Johnson: *Computers and Intractability - A Guide to the Theory of NP-Completeness* (W.H. Freeman and Co., San Francisco, 1979).

[Grötschel+Lovász+Schrijver81] M. Grötschel, L. Lovász and A. Schrijver: The elllipsoid method and its consequences in combinatorial optimization. *Combinatorica* **1** (1981) 169-197.

[Grötschel+Lovász+Schrijver88] M. Grötschel, L. Lovász and A. Schrijver: *Geometric Algorithms and Combinatorial Optimization* (Algorithms and Combinatorics **2**) (Springer, Berlin, 1988).

[Hajj81] I.N.Hajj, P.Yang and T.N.Trick: Avoiding zero pivots in the modified nodal approach. *IEEE Transactions on Circuits and Systems* **CAS-28** (1981) 271-279.

[Hall35] P. Hall: On representatives of subsets. *Journal of the London Mathematical Society* **10** (1935) 26-30.

[Harary69] *Graph Theory* (Addison-Wesley, Reading, MA, 1969).

[Hoffman+Kunze72] K.Hoffman and R.Kunze: *Linear Algebra* (2nd ed.) (Prentice-Hall International, inc., Englewood Cliffs, 1972). (Prentice-Hall of India, 1972).

[Hopcroft+Tarjan74] J. Hopcroft and R. Tarjan: Efficient planarity testing. *Journal of ACM* **21** (1974) 549-568.

[Horn55] A. Horn: A characterization of unions of linearly independent sets. *Journal of the London Mathematical Society* **30** (1955) 494-496.

[Imai83] H. Imai: Network-flow algorithms for lower truncated transversal polymatroids. *Journal of the Operations Research Society of Japan* **26** (1983) 186-211.

[Iri68] M. Iri: A min-max theorem for the ranks and term ranks of a class of matrices – an algebraic solution to the problem of the topological degrees of freedom of a network (in Japanese). *Transactions of the Institute of Electrical and Communication Engineers of Japan* **51A** (1968) 180-187.

[Iri69] M. Iri: The maximum rank minimum term rank theorem for the pivotal transformations of a matrix. *Linear Algebra and Its Applications* **2** (1969) 427-446.

[Iri71] M. Iri: Combinatorial canonical form of a matrix with applications to the principal partition of a graph (in Japanese). *Transactions of the Institute of Electronics and Communication Engineers of Japan* **54A** (1971) 30-37.

[Iri79a] M. Iri: Survey of recent trends in applications of matroids. *Proceedings of IEEE International Symposium on Circuits and Systems* Tokyo (1979) 987.

[Iri79b] M. Iri: A review of recent work in Japan on Principal partitions of matroids and their applications. *Annals of the New York Academy of Sciences* **319** (1979) 306-319.

[Iri83] M. Iri: Applications of matroid theory. In: *Mathematical Programming – The State of the Art* (A. Bachem, M. Grötschel and B. Korte, eds., Springer, Berlin, 1983) 158-201.

[Iri84] M. Iri: Structural theory for the combinatorial systems characterized by submodular functions. In: *Progress in Combinatorial Optimizations* (W.R. Pulleyblank, ed., Academic Press, Toronto, 1984), 197-219.

[Iri+Fujishige81] M. Iri and S. Fujishige: Use of matroid theory in operations research, circuits and system theory. *International Journal of Systems Science* **12** (1981) 27-54.

[Iri+Recski80] M. Iri and A. Recski: What does duality really mean? *International Journal of Circuit Theory and Its Applications* **8** (1980) 317-324.

[Iri+Tomi76] M. Iri and N. Tomizawa: An algorithm for finding an optimal 'independent assignment'. *Journal of the Operations Research Society of Japan* **19** (1976) 32-57.

[Iri+Tsunekawa+Murota82] M. Iri, J. Tsunekawa and K. Murota: Graph theoretical approach to large-scale systems – Structural solvability and block-triangularization. *Transactions of Information Processing Society of Japan* **23** (1982) 88-95.

[Iwata01] S Iwata, L Fleischer, S Fujishige: A Combinatorial Strongly Polynomial Algorithm for Minimizing Submodular Functions. *Journal of the ACM* **48** (2001) 761 - 777. Journal of the ACM, 2001

[Jacobson74] N.Jacobson: *Basic Algebra I&II*(W.H.Freeman and Co.,U.S.A,1974).

[Kamath94] M.V. Kamath: *Application of the partitioning approach to a class of routing problems..* Ph.D. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 1994.

[Kirchhoff1847] G.Kirchhoff: Ueber die Auflösung der Gleichungen, auf welche man bei der Untersuchungen der linearen Vertheilung galvanischer Ströme geführt wird. *Poggendorff Annalen der Physik und Chemie* **LXXII**, Leipzig, (1847) 497-508.

[Kishi+Kajitani68] G. Kishi and Y. Kajitani: Maximally distant trees in a linear graph (in Japanese). *Transactions of the Institute of Electronics and Communication Engineers of Japan* **51A** (1968) 196-203.

[Kishi+Kajitani69] G. Kishi and Y. Kajitani: Maximally Distant Trees and Principal Partition of a Linear Graph. *IEEE Transactions on Circuit Theory* **CT-16** (1969) 323-329.

[Kajitani+Sakurai+Okamoto] Y. Kajitani, Hidekazu Sakurai, Eiji Okamoto : Metric in the set of labelled graphs and its applications to network theory. preprint (undated), received (1982).

[König36] D.König: *Theorie der Endlichen und Unendlichen Graphen* (Leipsig, 1936, Reprinted New York, Chelsea, 1950).

[Kozen92] D.C.Kozen: *The Design and Analysis of Algorithms* (Springer-Verlag,New York, Berlin, London, Tokyo, 1992).

[Kron39] G. Kron: *Tensor Analysis of Networks* (J.Wiley, New York, 1939).

[Kron63] G. Kron: *Diakoptics - Piecewise Solution of Large Scale Systems* (McDonald,London,1963).

[Kuhn56] H.W. Kuhn: Solvability and consistency for linear equations and inequalities. *American Mathematical Monthly* **63** (1956) 217-232.

[Kung86] J.P.S. Kung: *A Source Book in Matroid Theory* (Birkhäuser, Boston, 1986).

[Laman70] G. Laman: On graphs and rigidity of plane skeletal structures. *Engineering Mathematics* **4** (1970) 331-340.

[Lawler76] E.L. Lawler: *Combinatorial Optimization – Networks and Matroids* (Holt, Rinehart and Winston, New York, 1976).

[Lovász83] L. Lovász: Submodular functions and convexity. In: *Mathematical Programmming – The State of the Art* (A.Bachem. M. Grötschel and B.Korte, eds., Springer, Berlin, 1983), 235-257.

[Lovász+Plummer86] L. Lovász and M. Plummer: *Matching Theory* (Akadémia Kiadó, Budapest; North-Holland, Amsterdam, 1986).

[MacDuffee33] C.C.MacDuffee: *The Theory of Matrices* (Springer, Berlin, 1933) (reprinted Chelsea, NewYork, 1946).

[MPM78] V.M.Malhotra, M.Pramod-Kumar and S.N.Maheshwari: An $O(V^3)$ algorithm for finding maximum flows in networks. *Information Processing Letters* **7** (1978) 277-278.

[Mason81] J. H. Mason: Glueing matroids together: a study of Dilworth truncations and matroid analogues of exterior and symmetric powers. In: *Algebraic Methods in Graph Theory.* Proceedings of the Colloquium, Szeged, 1978; (L.Lovász and V.T. Sós, eds., North-Holland Amserdam, 1981) 519-561.

[McCalla88] W.J.McCalla: *Fundamentals of computer-aided circuit simulation* (Kluwer Academic Publishers, Boston, 1988).

[McDiarmid73] C.J.H. McDiarmid: Independence structures and submodular functions. *Bulletin of the London Mathematical Society* **5** (1973) 18-20.

[McDiarmid75] C.J.H. McDiarmid: Rado's theorem for polymatroids. *Mathematical Proceedings of the Cambridge Philosophical Society* **78** (1975) 263-281.

[Menger27] K. Menger: Zur allgemeinen Kurventheorie. *Fundamenta Mathematicae* **10** (1927) 96-115.

[Minty60] G.J. Minty: Monotone Networks. *Proceedings of the Royal Society* Ser.A **257** (1960) 194-212.

[Minty66] G.J. Minty: On the axiomatic foundations of the theories of directed linear graphs, electrical networks and network-programming. *Journal of Mathematics and Mechanics* **15** (1966) 405-520.

[Mirsky71] L. Mirsky: *Transversal Theory* (Academic Press, London, 1971).

[Murota87] K. Murota: *Systems Analysis by Graphs and Matroids – Structural Solvability and Controllability* (Algorithms and Combinatorics **3**) (Springer, 1987).

[Murota88] K. Murota: Note on the universal bases of a pair of polymatroids. *Journal of the Operations Research Society of Japan* **31** (1988) 565-572.

[Murota90] K. Murota: Principal structure of layered mixed matrices. *Discrete Applied Mathematics* **27** (1990) 221-234.

[Murota+Iri85] K. Murota and M. Iri: Structural solvability of systems of equations – a mathematical formulation for distinguishing accurate and inaccurate numbers in structural analysis of systems. *Japan Journal of Applied Mathematics* **2** (1985) 247-271.

[Nagamochi+Ibaraki92a] H.Nagamochi and T.Ibaraki: Linear time algorithms for finding a sparse k-connected spanning subgraph of a k-connected graph.*Algorithmica* **7** (1992) 583-596.

[Nagamochi+Ibaraki92b] H.Nagamochi and T.Ibaraki: Computing edge connectivity in multigraphs and capacitated graphs. *SIAM Journal of Discrete Maths* **5** (1992) 54-66.

[Nagamochi+Ono+Ibaraki94] H.Nagamochi,T.Ono and T.Ibaraki: Implementing an efficient minimum capacity cut algorithm.*Mathematical Programming* **67**(1994) 325-341.

[Nakamura+Iri81] M. Nakamura and M. Iri: A structural theory for submodular functions, polymatroids and polymatoid intersections. Research Memorandum RMI 81-06, Department of Mathematical Engineering and Instrumentation Physics, Faculty of Engineering, University of Tokyo, August 1981.

[Narayanan74] H. Narayanan: *Theory of Matroids and Network Analysis* Ph.D. Thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, February 1974.

[Narayanan75] H. Narayanan: A topological formulation of Diakoptics and its applications. Research report, E.E. Department,I.I.T. Bombay, May 1975.

[Narayanan78] H. Narayanan: *A Topological Approach to Network Analysis* Monograph, Electrical Engineering Department, Indian Institute of Technology, Bombay, August 1978.

[Narayanan79] H. Narayanan: A theorem on graphs and its application to network analysis. *Proceedings of IEEE International Symposium on Circuits and Systems* (1979) 1008-1011.

[Narayanan80] H. Narayanan: A topological approach to network decomposition. Research Report, E.E. Department,I.I.T. Bombay, July 1980.

[Narayanan85a] H. Narayanan: A theorem on complementary orthogonal spaces and its generalizations. Research report, E.E. Department,I.I.T. Bombay, September 1985.

[Narayanan85b] H. Narayanan: Violations of port conditions in the interconnection of multiports. *International Journal of Circuit Theory and its Applications* **13** (1985) 358-361.

[Narayanan85c] H. Narayanan: On the equivalence of Minty's painting theorem and Tellegen's theorem. *International Journal of Circuit Theory and its Applications* **13** (1985) 353-357.

[Narayanan86a] H. Narayanan: On the decomposition of vector spaces. *Linear algebra and its applications* **76** (1986) 61-98.

[Narayanan86b] H. Narayanan: A unified construction of adjoint systems and networks. *International Journal of Circuit Theory and its Applications* **14** (1986) 263-276.

[Narayanan87] H. Narayanan: Topological transformations of electrical networks. *International Journal of Circuit Theory and its Applications* **15** (1987) 211-233.

[Narayanan90] H. Narayanan: On the minimum hybrid rank of a graph relative to a partition of its edges and its apppliction to electrical network analysis. *International Journal of Circuit Theory and its Applications* **18** (1990) 269-288.

[Narayanan91] H. Narayanan: The principal lattice of partitions of a submodular function. *Linear Algebra and its Applications* **144** (1991) 179-216.

[Narayanan95a] H. Narayanan: A rounding technique for the polymatroid membership problem. *Linear Algebra and its Applications* **221** (1995) 41-57.

[Narayanan95b] H. Narayanan: Convolution and Dilworth truncation of submodular functions. *Special Issue on Decision Sciences, Journal of Indian Institute of Science*, Bangalore **75** (1995) 25-47.

[Narayanan+Kamath94] H. Narayanan and M.V. Kamath: The principal lattice of partitions of the exclusivity function in bipartite graphs – theory and algorithms. Technical Report TR-151-94, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 1994.

[Narayanan+Vartak81] H. Narayanan and M.N. Vartak: An elementary approach to the principal partition of a matroid. *Transactions of the Institute of Electronics and Communication Engineers of Japan* **E64** (1981) 227-234.

[Narayanan+Roy+Patkar92] H. Narayanan, S. Roy and S. Patkar: Min k-cut and the principal partition of a graph. *Proceedings of Second National Seminar on Theoretical Computer Science* (India, 1992).

[Narayanan+Roy+Patkar96] H. Narayanan, S. Roy and S. Patkar: Approximation algorithms for min-k-overlap using the PLP approach. *Journal of Algorithms* **21** (1996) 306-330.

[Narsingh Deo74] Narsingh Deo: *Graph Theory with Applications to Engineering and Computer Science* (Prentice-Hall International Inc., Englewood Cliffs, 1974) Also (Prentice-Hall of India Private Limited, New Delhi, 1990).

[Nash-Williams61] C. St. J.A. Nash-Williams: Edge-disjoint spanning trees of finite graphs. *Journal of the London Mathematical Society* **36** (1961) 445-450.

[Nash-Williams64] C. St. J.A. Nash-Williams: Decomposition of finite graphs into forests. *Journal of the London Mathematical Society* **39** (1964).

[Nash-Williams67] C. St. J.A. Nash-Williams: An application of matroids to graph theory. In: *Theory of Graphs* Proceedings of the International Symposium, (Rome, 1966) (P. Rosenstiehl, ed., Gordon & Breach, New York, 1967) 263-265.

[Ohtsuki+Ishizaki+Watanabe68] T. Ohtsuki, Y. Ishizaki and H. Watanabe: Network analysis and topological degrees of freedom (in Japanese). *Transactions of the Institute of Electrical and Communication Engineers of Japan* **51A** (1968) 238-245.

[Ohtsuki+Ishizaki+Watanabe70] T.Ohtsuki, Y.Ishizaki and H.Watanabe: Topological Degrees of Freedom and Mixed Analysis of Electrical Networks. *IEEE Transactions on Circuit Theory* **CT-17** (1970) 491-499.

[Ore56] O. Ore: Studies on directed graphs, I. *Annals of Mathematics* **63** (1956) 383-405.

[Ovalekar+Narayanan92] V.S. Ovalekar and H. Narayanan: Fast loop matrix generation for hybrid analysis and a comparison of the sparsity of the loop impedance and MNA impedance submatrices. *Proceedings of IEEE International Symposium on Circuits and Systems* (1992).

[Ozawa74] T. Ozawa: Common trees and partitions of two-graphs (in Japanese). *Transactions of the Institute of Electronic and Communication Engineers of Japan* **57A** (1974) 383-390.

[Ozawa75] T. Ozawa: Solvability of linear electric networks. Memoirs of the Faculty of Engineering, Kyoto University **37** (1975) 299-315.

[Ozawa76] T. Ozawa: Topological conditions for the solvability of active linear networks. *International Journal of Circuit Theory and itsApplications* **4** (1976) 125-136.

[Ozawa+Kajitani79] T. Ozawa and Y.Kajitani: Diagnosability of Linear Active Networks. *IEEE Transactions on Circuits and Systems* **CAS-26** (1979) 485-489.

[Papadimitriou+Steiglitz82] C.H.Papadimitriou and K.Steiglitz: *Combinatorial Optimization - Algorithms and Complexity.* (Prentice Hall,Englewood Cliffs,N.J,1982).

[Patkar92] S. Patkar: *Study of structure of graphs through the principal lattice of partitions approach.* Ph.D. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 1992.

[Patkar+Narayanan91] S. Patkar and H. Narayanan: Fast algorithm for the principal partition of a graph. *Proceedings of Eleventh Annual Symposium on Foundations of Software Technology and Theoretical Computer Science* (FSTTCS-11) **LNCS-560** (1991) 288-306.

[Patkar+Narayanan92a] S. Patkar and H. Narayanan: Characterization theorems for the matroids arising in the principal lattice of partitions of the rank functions of a graph. Technical Report TR.073-92, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, February 1992.

[Patkar+Narayanan92b] S. Patkar and H. Narayanan: Principal lattice of partitions of submodular functions on graphs: Fast algorithm for principal partition and generic rigidity. *Proceedings of the Third Annual International Symposium on Algorithms and Computation* (Nagoya, Japan 1992).

[Patkar+Narayanan92c] S. Patkar and H. Narayanan: Fast sequential and randomized parallel algorithms for rigidity and approximate min-k-cut. *Proceedings of Twelfth Annual Symposium on FSTTCS* (New Delhi, 1992) (Springer Verlag, 1992).

[Perfect68] H. Perfect: Applications of Menger's graph theorem. *Journal of Mathematical Analysis and Applcations* **22** (1968) 96-111.

[Perfect69] H. Perfect: Independence spaces and combinatorial problems. *Proceedings of the London Mathematical Society* **19** (1969) 17-30.

[Pym+Perfect70] J.S.Pym and H.Perfect: Submodular functions and independence structures. *Journal of Mathematical Analysis and Applications,* **30** (1970) 1-31.

[Queyranne95] M.Queyranne: A combinatorial algorithm for minimizing symmetric submodular functions. *Proceedings of the Sixth Annual ACM-SIAM Symposium on Discrete Algorithms* (1995) 98-101.

[Rado42] R. Rado: A theorem on independence relations. *Quarterly Journal of Mathematics*, Oxford **13** (1942) 83-89.

[Randow76] R. von Randow: *Introduction to the Theory of Matroids* (Springer, Berlin, 1976).

[Recski89] A. Recski: *Matroid Theory and its Applications in Electric Network Theory and in Statics.* (Springer-Verlag,Berlin,Heidelberg,New York,London,Paris,Tokyo,1989).

[Recski+Iri80] A. Recski and M. Iri: Network theory and transversal matroids. *Discrete Applied Mathematics* **2** (1980) 311-326.

[Roy93] S. Roy: *The principal lattice of partitions approach to partitioning problems in VLSI*. Ph.D. Thesis, Department of Electrical Engineering, Indian Institute of Technology, Bombay, 1993.

[Roy+Narayanan91] S. Roy and H. Narayanan: A new approach to the problem of PLA partitioning using the theory of the principal lattice of partitions of a submodular function. *Proceedings of the Fourth Annual ASIC Conference and Exhibit* (Rochester, 1991).

[Roy+Narayanan93a] S. Roy and H. Narayanan: Application of the principal partition and principal lattice of partitions of a graph to the problem of decomposition of a finite state machine. *Proceedings of the IEEE International Symposium of Circuits and Systems* (Chicago, Illinois, 1993).

[Roy+Narayanan93b] S. Roy and H. Narayanan: An alternative derivation of the PLP based on a minimum cost rate theorem. Technical Report VLSI-93-2, VLSI Design Centre, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, February 1993.

[Roy+Gaitonde+Narayanan90] S. Roy, D. Gaitonde and H. Narayanan: BITSIM – A general purpose circuit simulator. *Journal of the Institute of Electrical and Telecommunication Engineers* **36** (1990) 265-273.

[Saran+Vazirani91] H.Saran and V.V. Vazirani: Finding a k-cut within twice the optimal. *Proceedings of the* 32^{nd} *Annual IEEE Symposium on the Foundations of Computer Science* (1991).

[Schrijver86] A. Schrijver: *Theory of Linear and Integer Programming* (John Wiley & Sons, New York, 1986).

[Schrijver00] A. Schrijver: A combinatorial algorithm minimizing submodular functions in strongly polynomial time. *Journal of Combinatorial Theory (B)*, **80** (2000), 346-365.

[Seshu+Reed61] S. Seshu and M.B. Reed: *Linear Graphs and Electrical Networks* (Addison-Wesley, Reading, Mass., London, 1961).

[Sleator80] D.D.Sleator: An $O(nm \log n)$ algorithm for maximum network flow. Technical Report STAN-CS-80-831, Stanford University (1980).

[Sleator+Tarjan 83] D.D.Sleator and R.E.Tarjan: Self-adjusting binary trees. *Proceedings of the 15th ACM Symposium on Theory of Computing* (1983) 235-245.

[Sohoni92] M. Sohoni: *The shape of polyhedra in combinatorial optimization.* Ph.D. Thesis, Department of Computer Science and Engineering, Indian Institute of Technology, Bombay, 1992.

[Stoer+Wagner94] M.Stoer and F.Wagner:A simple min cut algorithm.
Proceedings of the 1994 European Symposium on Algorithms **ESA94,LNCS855** (Springer, 1994) 141-147.

[Stoer+Witzgall70] J. Stoer and C. Witzgall: *Convexity and Optimization in Finite Dimensions* **I** (Springer, Berlin, 1970).

[Sugihara79] K. Sugihara: Studies on mathematical structures of line drawings of polyhedra and their applications to scene analysis. Research Electrotechnical Laboratory **800** (1979).

[Sugihara80] K. Sugihara: On redundant bracing in plane skeltal structures. *Bulletin of the Electrotechnical Laboratory* **44** (1980) 78-88.

[Sugihara82] K. Sugihara: Mathematical structures of line drawings of polyhedra: toward man-machine communication by means of line drawings. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **PAMI-4** (1982) 458-469.

[Sugihara83] K. Sugihara: A unifying approach to descriptive geometry and mechanisms. *Discrete Applied Mathematics* **5** (1983) 313-328.

[Sugihara84] K. Sugihara: An algebraic and combinatorial approach to the analysis of line drawings of polyhedra. *Discrete Applied Mathematics* **9** (1984) 77-104.

[Sugihara86] K. Sugihara: *Machine Interpretion of Line Drawings* (The MIT Press, Cambridge, Massachusetts, 1986).

[Sugihara+Iri80] K. Sugihara and M. Iri: A mathematical approach to the determination of the structure of concepts. *Matrix and Tensor Quarterly* **30** (1980) 62-75.

[Tarjan72] R.E. Tarjan: Depth-first search and linear graph algorithms. *SIAM Journal on Computing* **1** (1972) 146-160.

[Tomizawa75] N. Tomizawa: Irreducible matroids and classes of r-complete bases (in Japanese). *Transactions of the Institute of Electronics and Communication Engineers of Japan* **58A** (1975) 793-794.

[Tomizawa76] N. Tomizawa: Strongly irreducible matroids and principal partition of a matroid into irreducible minors (in Japanese). *Transactions of the Institute of Electronics and Communication Engineers of Japan* **59A** (1976) 83-91.

[Tomizawa80a] N. Tomizawa: Theory of hyperspace (I) – supermodular functions and generalization of concept of 'bases' (in Japanese). Papers of the Technical Group on Circuits and System Theory, Institute of Electronics and Communication Engineers of Japan CAS80-72 (1980).

[Tomizawa80b] N. Tomizawa: Theory of hyperspace (II) – geometry of intervals and superemodular functions of higher order. *ibid.*, CAS80-73 (1980).

[Tomizawa80c] N. Tomizawa: Theory of hyperspace (III) – Maximum deficiency = minimum residue theorem and its application (in Japanese). *ibid.*, CAS80-74 (1980).

[Tomizawa80d] N. Tomizawa: Theory of hyperspace (IV) – principal partitions of hypermatroids (in Japanese). *ibid.*, CAS80-85 (1980).

[Tomizawa+Fujishige82] N. Tomizawa and S. Fujishige: Historical survey of extensions of the concept of principal partition and their unifying generalization to hypermatroids. Systems Science Research Report No.5, Department of Systems Science, Tokyo Institute of Technology, April 1982 (also its abridgement in *Proceedings of the 1982 IEEE International Symposium on Circuits and Systems* (Rome, 1982) 142-145).

[Tomi+Iri74] N. Tomizawa and M. Iri: An algorithm for determining the rank of a triple matrix product AXB with applications to the problem of discerning the existence of the unique solution in a network (in Japanese). *Transactions of the Institute of Electronics and Communication Engineers of Japan* **57A** (1974) 834-841.

[Tutte58] W.T. Tutte: A homotopy theorem for matroids I,II. *Transactions of the American Mathematical Society* **88** (1958) 144-174.

[Tutte59] W.T. Tutte: Matroids and graphs. *Transactions of the American Mathematical Society* **90** (1959) 527-552.

[Tutte61] W.T. Tutte: On the problem of decomposing a graph into $n-$ connected factors. *Journal of the London Mathematical Society* **36** (1961) 221-230.

[Tutte65] W.T. Tutte: Lectures on matroids. *Journal of Research of the National Bureau of Standards* **B69** (1965) 1-48.

[Tutte71] W.T. Tutte: *Introduction to the Theory of Matroids* (American Elsevier, New York, 1971).

[Van Leeuwen90] J.Van Leeuwen (editor): *Handbook of Theoretical Computer Science (Volume A)- Algorithms and Complexity.* (Elsevier,Amsterdam and M.I.T Press,Cambridge,1990).

[Van der Waerden37] B.L. van der Waerden: *Moderne Algebra* (2nd ed.) (Springer, Berlin, 1937).

[von Neumann47] J. von Neumann: Discussion of a maximum problem. Unpublished working paper. Institute for Advanced Studies, Princeton,New Jersey (1947). Reprinted in
John von Neumann, Collected Works, vol VI. (Pergamon Press, Oxford, 1963).

[Welsh76] D.J.A. Welsh: *Matroid Theory* (Academic Press, Cambridge, 1976).

[White86] N.L. White: *Theory of Matroids* (Cambridge University Press, Cambridge, 1986).

[Whitney32] H. Whitney: Non-separable and planar graphs. *Transactions of the American Mathematical Society* **34** (1932) 339-362.

[Whitney33a] H. Whitney: Planar graphs. *Fundamenta Mathematicae* **21** (1933) 73-84.

[Whitney33b] H. Whitney: On the classification of graphs. *American Journal of Mathematics* **55** (1933) 236-244.

[Whitney33c] H. Whitney: 2-isomorphic graphs. *American Journal of Mathematics* **55** (1933) 245-254.

[Whitney35] H. Whitney: On the abstract properties of linear dependence. *American Journal of Mathematics* **57** (1935) 509-533.