Chapter 3

Elementary graph theory

3.1 Spanning forests and trees

An edge subgraph of G that has no undirected cycles is called a *forest* of G, and is called a *tree* of G if it is connected. A forest is a disjoint union of trees.

A forest F of G is a *spanning forest* if every pair of vertices that are connected in G are also connected in F. A spanning forest that is a tree is called a *spanning tree*.

Let F be a spanning forest of G. An edge of G is a *tree edge* (or *tree arc*) with respect to F if e belongs to F, and otherwise is a *nontree edge* (or arc).

Lemma 3.1.1. *If F is a spanning forest, $|E(F)| = |V(F)| - \kappa(F)$.*

Lemma 3.1.2. *Suppose F is a forest of G, and uv is an edge in $E(G) - E(F)$ such that u and v are not connected in F. Then $F \cup \{uv\}$ is a forest.*

Proof. Let $F' = F \cup \{uv\}$, and suppose F' has a simple cycle C. Then C must include the edge uv, for otherwise C is a cycle in F. But $C - \{uv\}$ is a path in F connecting u and v, a contradiction. □

We say an edge-subgraph F of G is a *spanning forest* if every pair u, v of vertices that are connected in G are also connected in F. Note that in this case $\kappa(F) = \kappa(G)$. If G is connected then a spanning forest is a tree, so we call it a *spanning tree* of G.

Corollary 3.1.3 (Matroid property of forests)**.** *For any forest F of G, there exists a set M of edges in $E(G) - F$ such that $F \cup M$ is a spanning forest of G.*

Corollary 3.1.4. *If F is a forest of G and $|E(F)| = |V(G)| - 1$ then F is a spanning tree of G.*

Proof. By Corollary 3.1.3, there exists a set M of edges in $E(G) - E(F)$ such that $F \cup M$ is a spanning forest of G. By Corollary 3.1.1,

$$\begin{aligned} |V(G)| - \kappa(G) &= |F| + |M| \\ &= |V(G)| - 1 + |M| \end{aligned}$$

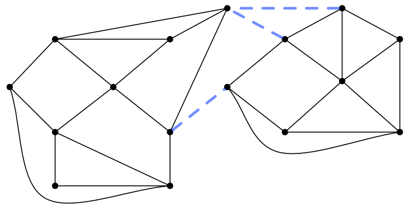

Figure 3.1: A graph is shown; the dashed edges form a cut.

so $1-|M| = \kappa(G)$. Since $\kappa(G) \geq 1$, we can infer that $|M| = 0$ and $\kappa(G) = 1$. □

3.1.1 Nontree edges and fundamental cycles

Let G be a graph, and let F be a spanning forest of G. For a dart d of an nontree edge, there is a simple head(d)-to-tail(d) path $d_1 \cdots d_k$ of darts in G whose edges belong to F. Write $d_0 = d$ so $d_0 \cdots d_k$ is a simple cycle C_e of darts, called the *fundamental cycle of* d *with respect to* F. For an arc e of $E(G) - F$, we define the fundamental cycle of e to be the fundamental cycle of the primary dart $(e, +1)$.

3.2 Cuts

3.2.1 (Undirected) cuts

For a graph G and a set S of vertices of G, we define $\delta_G(S)$ to be the set of edges having one endpoint in S and one endpoint not in S. We say a set of edges of G is a *cut* of G if it has the form $\delta_G(S)$.

3.2.2 (Directed) dicuts

We define $\delta_G^+(S)$ to be the set of arcs whose tails are in S and whose heads are not in S. Note that $\delta_G^+(S)$ is a subset of $\delta_G(S)$. A set of arcs of G is a *directed cut* (a.k.a. *dicut*) if it has the form $\delta_G^+(S)$.

3.2.3 Dart cuts

For a set S of vertices of G, we define $\vec{\delta}_G(S)$ to be the set of *darts* whose tails are in S and whose heads are not in S. Note that $\vec{\delta}_G(S)$ has one dart for each edge in $\delta_G(S)$. We refer to $\vec{\delta}_G(S)$ as the *dart boundary of* S *in* G.

3.2.4 Bonds/simple cuts

Let G be a graph, let K be a connected component of G, and let S be a subset of the vertices of K. We say a cut $\delta_G(S)$ or a dart cut $\vec{\delta}_G(S)$ is a *bond* or a *simple cut* if S is connected in G and $V(K) - S$ is connected in G.

Note that a cut in G is minimal among nonempty cuts iff it is a simple cut. Any cut can be written as a disjoint union of simple cuts.

Vertex cuts Now let S be a set of edges. We define $\partial_G(S)$ to be the set of vertices v such that at least one edge incident to v is in S and at least one edge incident to v is *not* in S. We refer to $\partial_G(S)$ as the *vertex boundary of S in G*. A vertex of $\partial_G(S)$ is a *boundary vertex* of S in G.

Notational conventions We may omit the subscript and write $\delta(S)$ or $\delta^+(S)$ when doing so introduces no ambiguity.

For a vertex v, we may write $\delta(v)$ or $\vec{\delta}(v)$ to mean $\delta(\{v\})$ or $\vec{\delta}(\{v\})$.

Question 3.2.1. *Give a graph G and a vertex v for which $\vec{\delta}_G(v)$ is not identical to the set v of darts.*

3.2.5 Tree edges and fundamental cuts

Let G be a graph, and let F be a spanning forest of G.

For a tree edge $e = u_1u_2$ (where $u_1 = \text{tail}(e)$ and $u_2 = \text{head}(e)$), let K be the connected component of F that contains e. For $i = 1, 2$, let

$$S_i = \{\text{vertices reachable from } u_i \text{ via edges of } F - e\}$$

Claim: S_1 and S_2 form a partition of the vertices of K.

Proof. Let T be the tree connecting the vertices of F. For any vertex v of K, for $i = 1, 2$, let P_i be the simple v-to-u_i path in T. If e were in neither P_1 nor P_2 then $P_1 \circ \text{rev}(P_2) \circ e$ would be a simple cycle in T, so e is in one of them, say P_1. Then the prefix of P_1 ending just before e is a simple v-to-u_2 path not using e, so v is in S_2. □

We call $\vec{\delta}_G(S_1)$ the *fundamental cut of e in G with respect to F*. Since S_1 and S_2 are connected in K, the claim implies the following.

Lemma 3.2.2 (Fundamental-Cut Lemma)**.** *For any tree edge e, the fundamental cut of e is a simple cut.*

Lemma 3.2.3. *For distinct tree edges e, e', e' is not in the fundamental cut of e.*

Problem 3.2.4. *Prove Lemma 3.2.3.*

3.2.6 Paths and Cuts

Lemma 3.2.5 (Path/Cut Lemma)**.** *Let G be a graph, and let u and v be vertices of G.*

- Dipath/Dicut *For a set A of arcs, every u-to-v dipath contains an arc of A iff there is a dicut $\delta^+(S) \subseteq A$ such that $u \in S, v \notin S$.*

- Path/Cut *For a set E of edges, every u-to-v path contains an edge of A iff there is a cut $\delta(S) \subseteq A$ such that $u \in S, v \notin S$.*
- Dart Path/Dart Cut *For a set D of darts, every u-to-v path of darts contains a dart of D iff there is a dart cut $\vec{\delta}(S) \subseteq D$ such that $u \in S, v \notin S$.*

Proof. We give the proof for the first statement; the proofs for the others are similar.

Let S be the set of vertices reachable from x via paths that avoid arcs in A.

(*only if*) Suppose every x-to-y dipath contains an arc of A. Then $x \in S, y \notin S$. Let uv be an arc in $\delta^+(S)$. then $u \in S$, so there is an x-to-u path P that avoids arcs in A. On the other hand, $v \notin S$, so every x-to-v path contains an arc in A. Consider the path $P\ uv$. It is an x-to-v path, so contains some arc in A, but P has no arcs of A, so $uv \in A$.

(*if*) Suppose there is a dicut $\delta^+(S) \subseteq A$ such that $x \in S, y \notin S$. Let P be any x-to-y path, and let v be the first vertex in P that does not belong to S (there is at least one such vertex, since $y \notin S$). Let u be the predecessor of v in P. By choice of v, we know $u \in S$. Hence $uv \in \delta^+(S)$, so $uv \in A$. □

3.2.7 Two-edge-connectivity and cut-edges

We define an equivalence relation, two-edge-connectivity, on edges. Edges e_1 and e_2 are *two-edge-connected* in G if G contains a cycle of edges containing both of them. The equivalence classes of this relation are called *two-edge-connected components.*

An edge e of G is a *cut-edge* if the two-edge-connected component containing e contains no other edges.

Lemma 3.2.6 (Cut-Edge Lemma)**.** *An edge e of G is a cut-edge iff every path between its endpoints uses e.*

3.3 Vector Spaces

Dart space Let $G = (V, E)$ be a graph. The *dart space* of G is $\mathbb{R}^{E\times\{\pm 1\}}$, the set of vectors $\boldsymbol{\alpha}$ that assign a real number $\boldsymbol{\alpha}[d]$ to each dart d. For a vector $\boldsymbol{c}$ in dart space and given a set S of darts, $\boldsymbol{c}(S)$ denotes $\sum_{d\in S} \boldsymbol{c}[d]$.

Vertex space The *vertex space* of G is $\mathbb{R}^V$. A vector of vertex space is called a *vertex vector.*

Arc space and arc vectors The *arc space* of G is a vector subspace of the dart space, namely the set of vectors $\boldsymbol{\alpha}$ in the dart space that satisfy *antisymmetry*:

$$\text{for every dart } d, \boldsymbol{\alpha}[d] = -\boldsymbol{\alpha}[\text{rev}(d)] \tag{3.1}$$

A vector in arc space is called an *arc vector*. We will mostly be working with arc vectors.

$\boldsymbol{\eta}(d)$ For a dart d, define $\boldsymbol{\eta}(d)$ to be the arc vector such that $\boldsymbol{\eta}(d)[d] = 1$ and $\boldsymbol{\eta}(d)[d'] = 0$ for all darts d' such that $d' \neq d$ and $d' \neq \text{rev}(d)$.

Fact 3.3.1. *The vectors* $\{\boldsymbol{\eta}((a, +1)) \ : \ a \in E\}$ *form a basis for the arc space.*

We extend this notation to sets of darts: $\boldsymbol{\eta}(S) = \sum_{d \in S} \boldsymbol{\eta}(d)$. Formally, a vertex v is the set of darts having v as tail, so $\boldsymbol{\eta}(v) = \sum_d \boldsymbol{\eta}(d)$ where the sum is over those darts whose tails are v.

For a dart d,

$$\boldsymbol{\eta}(v)[d] = \begin{cases} 1 & \text{if only } a\text{'s tail is } v \\ -1 & \text{if only } a\text{'s head is } v \\ 0 & \text{otherwise} \end{cases}$$

A self-loop cancels itself out.

The dart-vertex incidence matrix A_G For a graph G, we denote by A_G the *dart-vertex incidence matrix*, the matrix whose columns are the vectors $\{\boldsymbol{\eta}(v) \ : \ v \in V(G)\}$. That is, A_G has a row for each dart d and a column for each vertex v, and the dv entry is 1 if v is the head of d, -1 if v is the tail of d, and zero otherwise.

3.3.1 The cut space

Let G be a graph. The vector space spanned by the set $\{\boldsymbol{\eta}(\vec{\delta}_G(v)) \ : \ v \in V\}$ is called the *cut space* of G. To define a basis for this vector space, let $K_1, \ldots, K_{\kappa(G)}$ be the connected components of G, and let $v_1, \ldots, v_{\kappa(G)}$ be representative vertices chosen arbitrarily from the vertex sets of the components. Let $\text{CUT}_G = \{\boldsymbol{\eta}(\vec{\delta}_G(v)) \ : \ v \in V - \{\hat{v}_1, \ldots, \hat{v}_{\kappa(G)}\}\}$. Note that $|\text{CUT}_G| = |V| - \kappa(G)$. Clearly each vector in CUT_G belongs to the cut space. We will eventually prove that CUT_G is a basis for the cut space. (For brevity, we may omit the subscript when the choice of graph G is clear.)

Lemma 3.3.2. *The vectors in CUT are linearly independent, so span(CUT) has dimension |CUT|.*

Proof. Suppose $\boldsymbol{\psi} = \sum_v \psi_v \boldsymbol{\eta}(v)$ is a nonzero linear combination of vectors in CUT. We show that the sum is not the all-zeroes vector. Let H be the subgraph induced by the set of vertices v such that $\psi_v \neq 0$. Let K be a connected component of H. Since K is a proper subgraph of some connected component K' of G itself (K includes no representative vertex $\hat{v}_i$), there is some arc uv having exactly one endpoint in K. Assume without loss of generality that u belongs to K. Its other endpoint v cannot lie in another component of H, else u and v would be in the same component. Hence $\psi_v = 0$. This implies that the component of $\boldsymbol{\psi}$ corresponding to uv is nonzero. □

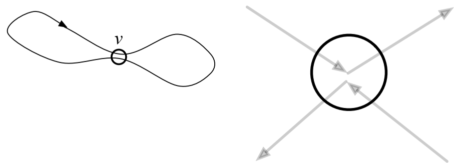

Figure 3.2: Consider a walk W and a vertex v. The darts of W that enter v are matched up with darts of W that leave v. In the vector $\boldsymbol{\eta}(v)$, darts leaving v are represented by!+1, and darts entering v are represented by -1, so $\boldsymbol{\eta}(v) \cdot \boldsymbol{\eta}(W) = 0$.

3.3.2 The cycle space

We turn to another vector space. We define the *cycle space* of G to be the orthogonal complement of the cut space in the arc space. That is, the cycle space is

$$\{\boldsymbol{\theta} \in \text{arc space} \ : \ \boldsymbol{\theta} \cdot \boldsymbol{\eta}(v) = 0 \text{ for all } v \in V\} \tag{3.2}$$

It follows via elementary linear algebra that the dimension of the cycle space plus the dimension of the cut space is $|E|$.

To define a basis for the cycle space, consider G as an undirected graph, and let F be a spanning forest. For each arc e in $E(G) - F$ (i.e., each nontree arc), we use C_e to denote the fundamental cycle of e with respect to F (defined in Section 3.1.1). We define $\boldsymbol{\beta}_F(e)$ to be $\boldsymbol{\eta}(C_e)$. We may omit the subscript F when it is clear which forest is intended.

We will show that the vectors in the set $\text{Cyc}_F = \{\boldsymbol{\beta}_F(e) \ : \ e \in E - F\}$ are independent and belong to the cycle space. Note that $|\text{Cyc}_F| = |E| - |F|$.

Lemma 3.3.3. *The vectors in* Cyc_F *are independent, so span*(Cyc_F) *has dimension* $|\text{Cyc}_F|$.

Proof. For any $e \in E - F$, the only vector in Cyc_F with a nonzero entry in the position corresponding to e is $\boldsymbol{\beta}_F(e)$, so this vector cannot be written as a linear combination of other vectors in Cyc_F. □

The following lemma partially explains the name of the cycle space.

Lemma 3.3.4. *If* W *is a closed walk then* $\boldsymbol{\eta}(W)$ *is in the cycle space.*

The proof is illustrated in Figure 3.2.

Proof. Let v be a vertex. For each dart d in W whose tail is v, v is the head of the predecessor of d in W, and for each dart d in W whose head is v, v is the tail of the successor of d. This shows that the number of darts of W whose tail is v equals the number of darts of W whose head is v, proving $\boldsymbol{\eta}(v) \cdot \boldsymbol{\eta}(W) = 0$. This shows that $\boldsymbol{\eta}(W)$ lies in the cycle space as defined in 3.2. □

Corollary 3.3.5. *The vectors in* $\textsc{Cyc}_F$ *belong to the cycle space so span*($\textsc{Cyc}_F$) *has dimension at most that of the cycle space.*

Proof. Every vector $\boldsymbol{\beta}_F(e)$ in $\textsc{Cyc}_F$ is equal to $\boldsymbol{\eta}(C_e)$ where C_e is a cycle, so by Lemma 3.3.4 belongs to the cycle space. □

3.3.3 Bases for the cut space and the cycle space

Now we put the pieces together.

Corollary 3.3.6. *CUT is a basis for the cut space, and* $\textsc{Cyc}_F$ *is a basis for the cycle space.*

Proof. By Lemma 3.3.2, for some nonnegative integer j_1,

$$\begin{aligned} \text{dim(cut space)} &= j_1 + |\text{CUT}| && (3.3)\\ &= j_1 + |V| - \kappa(G) && (3.4) \end{aligned}$$

By Corollary 3.3.5 and Lemma 3.3.3, for some nonnegative integer j_2,

$$\begin{aligned} \text{dim(cycle space)} &= j_2 + |\textsc{Cyc}_F| && (3.5)\\ &= j_2 + |E| - |F| && (3.6) \end{aligned}$$

Since the cut space and the cut space are orthogonal,

$$\begin{aligned} |E| &= \text{dim(arc space)} && (3.7)\\ &= \text{dim(cut space)} + \text{dim(cycle space)} && (3.8)\\ &= j_1 + |V| - \kappa(G) + j_2 + |E| - |F| && (3.9)\\ &= j_1 + |V| - \kappa(G) + j_2 + |E| - (|V| - \kappa(G)) && (3.10)\\ &= j_1 + j_2 + |E| && (3.11) \end{aligned}$$

so $j_1 = 0$ and $j_2 = 0$, proving that CUT is a basis for the cut space and $\textsc{Cyc}_F$ is a basis for the cycle space. □

We call $\textsc{Cyc}_F$ the *fundamental-cycle basis with respect to* F.

3.3.4 Another basis for the cut space

Let F be a spanning forest of G. Consider the set of vectors

$$\{\boldsymbol{\eta}(\text{fundamental cut of } e \text{ with respect to } F)\ : e \in F\}$$

Clearly each vector is in the cut space. Since distinct tree edges are not in each other's fundamental cuts (Lemma 3.2.3), these vectors are linearly independent. The set consists of $|F|$ vectors. Since F is a spanning forest of G, $|F| = |V(G)| - \kappa(G)$. The set of vectors therefore has the same cardinality as |CUT|. It follows that this set of vectors is another basis for the cut space, and we call it the *fundamental-cut basis* with respect to F.

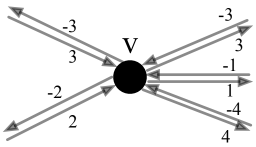

Figure 3.3: The two darts that enter v from its left carry positive amounts (3 and 2) of the commodity. The three darts that leave v to its right carry positive amounts (3, 1, and 4). The net outflow is $3 + 1 + 4 - (3 + 2) = 3$.Because of antisymmetry, a simpler way to calculate net outflow at v is to sum the values assigned to all darts leaving v: $(-3) + (-2) + 3 + 1 + 4 = 3$.

3.3.5 Conservation and circulations

Let $\boldsymbol{\gamma}$ be an arc vector. We can interpret $\boldsymbol{\gamma}$ as a plan for transporting amounts of some commodity (e.g. oil) along darts of the graph. If $\boldsymbol{f}[d] > 0$ for some dart d then we think of $\boldsymbol{\gamma}[d]$ units of the commodity being routed along dart d. Because $\boldsymbol{\gamma}$ satisfies antisymmetry (3.1), $\boldsymbol{\gamma}[\text{rev}(d)] < 0$ in this case.

The *(net) outflow* of $\boldsymbol{\gamma}$ at a vertex v is defined to be $\sum\{\boldsymbol{\gamma}[d] \ : \ d \in \vec{\delta}(v)\}$ This is the net amount of the commodity that leaves v (see Figure 3.3).

We say $\boldsymbol{\gamma}$ satisfies *conservation* at v if the net outflow at v is zero.

It follows from (3.2) that an arc vector that satisfies conservation at every vertex belongs to the cycle space. A vector $\boldsymbol{\theta}$ in the cycle space of G is called a *circulation* of G. We can interpret a circulation as a plan for transporting a commodity through the graph in such a way that no amount is created or consumed at any vertex. Circulations will play an essential role in our study of max-flow algorithms for planar graphs.

3.4 Embedded graphs

In solving the problem of the bridges of Königsberg, Euler discovered the power of abstracting a topological structure, a graph, from an apparently geometric structure. Perhaps it was this experience that enabled him to make another discovery. Polyhedra had been studied in ancient times but nobody seems to have noticed that every three-dimensional polyhedron without holes obeyed a simple relation:

$$n - m + \phi = 2$$

where n is the number of vertices, m is the number of edges, and ϕ is the number of faces. This equation is known as Euler's formula.[1]

This equation does not describe the geometry of a polyhedron; in fact, one can stretch and twist a polyhedron, and the formula will remain true (though the edges and faces will get distorted). We presume it was Euler's ability to think *beyond* the geometry that enabled him to realize the truth of this formula.

[1]There is another "Euler's formula," $e^{ix} = \cos x + i \sin x$.

Planar embedded graphs are essentially the mathematical abstraction of our stretched and twisted polyhedra. Turning Euler's observation on its head, we will end up defining planar embedded graphs as those embedded graphs that satisfy Euler's formula. The traditional definition of planar embedded graphs is geometric. Our definition of embedded graphs will not involve geometry at all. Instead, we will use the notion of a *combinatorial embedding.* The advantage of this approach is that it's easier to prove things about combinatorial embeddings. For that, we need to review permutations.

Permutations For a finite set S, a *permutation* on S is a function $\pi : S \longrightarrow S$ that is one-to-one and onto. That is, the inverse of π is a function. A permutation π on S is a *cycle* (sometimes called a *cyclic permutation*) if S can be written as $\{a_0, a_1, \ldots, a_{k-1}\}$ such that $\pi(a_i) = a_{(i+1) \bmod k}$ for all $0 \leq i < k$. According to the traditional notation for a cyclic permutation, we would then write π as $(a_0\ a_1\ \ldots\ a_{k-1})$. This notation is not unique; for example, $(a_1\ a_2\ \ldots\ a_{k-1}\ a_0)$ represents the same permutation. The length of the cycle is defined to be k.

Orbits The *orbits* of a permutation π are the equivalence classes under the equivalence relation where $c \sim d$ if there exists k such that $\pi^k(c) = d$. Here the exponent indicates k-fold composition, so $c \sim d$ if one can get from c to d by some number of applications of π.

Decomposition of a permutation into cyclic permutations For any orbit of a permutation π, the restriction of π to that orbit is a cyclic permutation. It follows that any permutation can be decomposed uniquely into nonempty cyclic permutations.

3.4.1 Embeddings

The idea of a combinatorial embedding was implicit in the work of Heffter (1891). Edmonds (1960) first made the idea explicit in his Master's thesis, and Youngs (1963) formalized the idea. A combinatorial embedding is sometimes called a *rotation system.*

Here's the idea. Suppose we start with an embedding of a graph in the plane. For each vertex, walk around the vertex counterclockwise and you will encounter edges in some order, ending up in the same place you started. The embedding thus defines a cyclic permutation (called a *rotation*) of the edges incident to the vertex. There is a sort of converse: given a rotation for each vertex, there is an embedding on *some* orientable surface that is consistent with those rotations.

Embedding For a graph $G = (V, E)$, an *embedding* of G is a permutation π of the set of darts $E \times \{\pm 1\}$ whose orbits are exactly the parts of V. Thus π

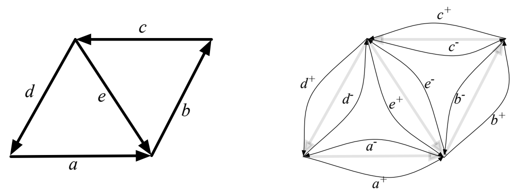

Figure 3.4: An embedded graph is illustrated. The permutation cycle associated with the top-left vertex is $(d^- \; e^- \; c^+)$, and the one associated with the bottom-right vertex is $(b^- \; e^+ \; a^+)$. This drawing reflects the convention that the cycle associated with a vertex represents the *counterclockwise* arrangement of darts around the vertex. In the drawing on the right, the individual darts are shown. This drawing is in accordance with the US "rules of the road": if the two darts are interpreted as two lanes of traffic, one travels in the right lane.

assigns a cyclic permutation to each part of V, i.e. each vertex. For each vertex v, we define $\pi|v$ to be the cyclic permutation that is the restriction of π to v.

> Our *interpretation* is that $\pi|v$ tells us the arrangement of darts counterclockwise around v in the embedding. Such an interpretation is useful in drawings of embedded graphs, e.g. the drawing on the left of Figure 3.4.

> The drawing on the right of Figure 3.4 should not be considered a drawing of an embedded graph. This drawing shows the

We use $V(\pi)$ to denote the set of orbits of π.

Embedded graph We define the pair $G = (\pi, E)$ to be an *embedded graph.* To be consistent with the definitions of unembedded graphs, we define $E((\pi, E)) = E$ and $V((\pi, E)) = V(\pi)$. We also define $\pi((\pi, E)) = \pi$. The *underlying graph* of an embedded graph G is defined to be the (unembedded) graph $(V(G), E(G))$.

$\vec{\delta}_G(S)$ For an embedded graph G and a set S of vertices, $\vec{\delta}_G(S)$ is a permutation on the set of darts whose tails are in S and whose heads are not in S....

Faces To define the faces of the embedded graph, we define another permutation dual(π) of the set of darts by composing rev with π :

$$\text{dual}(\pi) = \text{rev} \circ \pi$$

Then the *faces* of the embedded graph (π, E) are defined to be the orbits of dual(π). We typically denote dual(π) by π^*.

Consider, for example, the embedded graph of Figure 3.4.

$$\begin{aligned}
\pi^*[a^-] &= \text{rev}(\pi[a^-]) \\
&= \text{rev}(d^+) \\
&= d^- \\
\pi^*[d^-] &= e^+ \\
\pi^*[e^+] &= a^-
\end{aligned}$$

so one of the faces is $\{a^-, d^-, e^+\}$.

Note that, in the figure, the face's cyclic permutation of darts traces out a clockwise cycle of darts such that no edges are embedded within the cycle. Consider, though, the face consisting of $\{a^+, b^+, c^+, d^+\}$. In the drawing, the darts of this face appear to form a counterclockwise cycle, and the rest of the graph is embedded within this cycle. According to traditional nomenclature for planar embedded graphs, this face is called the *infinite face* because the edge-free region it bounds is infinite. However, imagine the same embedding on the surface of a sphere; there is no infinite face. All faces have equal status.

Since the combinatorial definition of embedded graphs and faces does not distinguish any faces, it is often convenient to imagine that it describes an embedding on a sphere (or other closed, orientable surface). However, for some purposes, it is convenient to distinguish one face, and designate it as the infinite face. Any face of the embedded graph can be chosen to be the infinite face, and this freedom is exploited in the design of some algorithms.

Remark: *In the traditional, geometric definition of embedded graphs, one considers the set of points that are* not *in the image of the embedding; a face is a connected component of this set of points. This definition works for connected graphs. However, for disconnected graphs, this definition leads to a face being in a sense shared by two components of the graph. Such a face has a disconnected boundary. This is a flawed definition; we later mention a couple of bad consequences of adopting this definition.*

3.4.2 Euler characteristic and genus

Let n, m, and ϕ be the number of vertices, edges, and faces of an embedded graph. The *Euler characteristic* of the embedding is $n - m + \phi$. The *genus* of the embedding is the integer g that satisfies the formula

$$n - m + \phi = 2 - 2g$$

As we discuss in Chapter 4, an embedding is planar if its genus is 0

3.4.3 Remark on the connection to geometric embeddings

From a combinatorial embedding, one can construct a surface and an embedding of the underlying graph in that surface. For each face $(d_1 \ \ldots \ d_r)$, construct

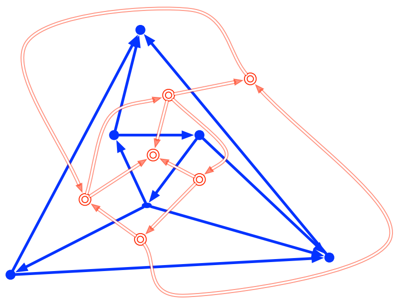

Figure 3.5: The primal and the dual are shown together. The arcs and vertices of the primal are drawn thick and solid and blue; the arcs and vertices of the dual are drawn with thin double lines in red.

an r-sided polygon and label the sides $d_1, \ldots, d_r$ in clockwise order. Now we have one polygon per face. For each edge e, glue together the two polygon sides labeled with the two darts of e. The result can be shown to be a closed, orientable surface. The graph is embedded in it.

Conversely, given any embedding of a graph G onto an closed, orientable surface, define π by the rotations at the vertices. Then the embedding defined by the gluing construction is homeomorphic to the given embedding. Thus, up to homeomorphism the rotations determine the embedding. The proof of this theorem is very involved, and won't be covered here. Since for the purpose for proofs we use combinatorial embeddings rather than geometric embeddings, the theorem will not be needed.

3.4.4 The dual graph

For an embedded graph $G = (\pi, E)$, the *dual graph* (or just *dual*) is defined to be the embedded graph $\text{dual}(G) = (\text{dual}(\pi), E)$. We typically denote the dual of G as G^*.

In relation to the dual, the original graph G is called the *primal graph* (or just the *primal*). Note that the vertices of G^* are the faces of G. It follows from the following lemma that the faces of G^* are the vertices of G.

Lemma 3.4.1 (The dual of the dual is the primal.)**.** *dual*(*dual*(G)) $= G$.

Problem 3.4.2. *Prove Lemma 3.4.1.*

Formally, there is no need to say more about the dual. Each orbit of π^* is a subset of darts and so can be interpreted as a vertex, so (π^*, E) is an embedded graph. However, for the sake of intuition, it is often helpful to draw the dual of an embedded graph superimposed on a drawing of the primal, as shown in Figure 3.5. Each dual vertex is drawn in the middle of a face of the primal, and, for each arc a of the primal, the corresponding arc of the dual is drawn so that it crosses a at roughly a right angle.
We often adopt the convention of drawing $G = (\pi, E)$ in such a way that the counterclockwise order of darts about a vertex corresponds to their order in the corresponding permutation cycle of π. That is, for a dart d with tail v, the next counterclockwise dart with tail v is $\pi[d]$.
However, when we draw the dual superimposed on the primal as we have described, the ordering of darts in a permutation cycle corresponds in the drawing to a *clockwise* arrangement.

Face vectors Because the faces of G are the vertices of the dual G^*, a vertex vector of G^* is an assignment of numbers to the *faces* of G. We therefore refer to it as a *face vector* of G.

3.4.5 Connectedness properties of embedded graphs

Lemma 3.4.3. *For any face f of any embedded graph G, the darts comprising f are connected.*

Proof. Let d and d' be darts of f. $\pi^*|f = (d_0\ d_1\ \ldots\ d_{k-1})$ where $d_0 = d$. Suppose $d_i = d'$. We claim that $d_0\ d_1\ d_2\ \ldots\ d_i$ is a walk in G, which proves the lemma. For $j = 1, 2, \ldots, i$, we have $d_j = \pi^*(d_{j-1})$. By definition of π^*, $\text{rev}(d_j) = \pi(d_{j-1})$, so $d_{j-1})$ and $\text{rev}(d_j)$ have the same head in G. Thus $\text{head}_G(d_{j-1}) = \text{tail}_G(d_j)$. □

Corollary 3.4.4. *If d and d' are connected in G then they are connected in G^*.*

Proof. Let $d_0\ d_1\ d_2\ \ldots\ d_k$ be a walk in G. For $i = 1, 2, \ldots, k$, $\text{head}_G(d_{i-1}) = \text{tail}_G(d_i)$, so $\text{tail}_G(\text{rev}(d_{i-1}) = \text{tail}_G(d_i)$, so $\text{rev}(d_{i-1})$ and d_i are in the same orbit of π. Hence $\text{rev}(d_{i-1})$ and d_i belong to the same face of π^*. By Lemma 3.4.3, $\text{rev}(d_{i-1})$ and d_i are connected in G^*, so d_{i-1} and d_i are connected in G^*. □

Corollary 3.4.5 (Connectivity Corollary)**.** *For any embedded graph G, a set of darts forms a connected component of G iff the same set forms a connected component in G^*.*

3.4.6 Cut-edges and self-loops

Lemma 3.4.6. *If e is not a self-loop in an embedded graph G then e is not a cut-edge in G^*.*

Proof. Let f be a face of G^* containing one of the two darts of e. Since e is not a self-loop in G, f does not contain the other dart of e. Therefore e is not a cut-edge in G^*. □

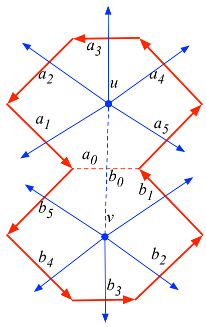

Figure 3.6: The primal graph G is shown in blue, and the dual G^* is shown in red. The edge e to be deleted from G^* has two darts, a_0 and b_0, which are in different faces of the dual G^*. When e is deleted, the two faces merge into one.

We shall show later that the converse of Lemma 3.4.6 holds in *planar* embedded graphs. However, more generally....

Problem 3.4.7. *Show that the converse of Lemma 3.4.6 does not hold.*

3.4.7 Deletion

Deleting a dart $\hat{d}$ from a permutation π of E is obtaining the permutation π' of $E - \{\hat{d}\}$ defined as follows.

$$\pi'[d] = \begin{cases} \pi[\pi[d]] & \text{if } \pi[d] = \hat{d} \\ \pi(d) & \text{otherwise} \end{cases}$$

Deleting an edge $\hat{e}$ consists of deleting the two darts of $\hat{e}$ (in either order).

Let π' be the permutation obtained from π by deleting an edge $\hat{e}$, and let $G' = (\pi', E - \{\hat{e}\})$ be the corresponding embedded graph. It is easy to check that the orbits of π' are the same as the orbits of π except that darts of $\hat{e}$ have been removed (possibly some orbits go away). Hence the underlying graph of G' is the graph obtained by deleting $\hat{e}$ from the underlying graph of G. We write $G - \hat{e}$ for the embedded graph obtained by deleting $\hat{e}$.

3.4.8 Compression (deletion in the dual) and contraction

Lemma 3.4.8. *For an embedded graph G, if e is not a self-loop then the underlying graph of dual$(G^* - e)$ is the graph obtained from the underlying graph of G by contracting e.*

Proof. The proof is illustrated in Figure 3.6. Let u and v be the endpoints of e. Let $a_0, \ldots, a_k$ be the darts outgoing from u, and let $b_0, \ldots, b_\ell$ be the darts outgoing from v, where a_0 and b_0 are the darts of e. Since e is not a self-loop, a_0 does not occur among the b_i's and b_0 does not occur among the a_i's.

In G^*, u is a face with boundary $(a_0\ a_1\ \cdots\ a_k)$ and v is a face with boundary $(b_0\ b_1\ \cdots\ b_\ell)$. Let $G^{*\prime} = (\pi^{*\prime}, E - \{e\})$ be the graph obtained from G^* by deleting e.

$$\begin{aligned}\text{dual}(\pi^{*\prime})[d] &= \pi^{*\prime} \circ \text{rev}(d) \\ &= \begin{cases} \pi^*[\pi^*[\text{rev}(d)]] & \text{if } \pi^*[\text{rev}(d)] \text{ is deleted} \\ \pi[d] & \text{otherwise} \end{cases} \qquad (3.12)\end{aligned}$$

For which two darts d is $\pi^*[\text{rev}(d)]$ deleted? Since $\pi^*[\text{rev}(d)] = \pi[d]$, the two darts are $\pi^{-1}[a_0]$, which is a_k, and $\pi^{-1}[b_0]$, which is b_ℓ.

Rewriting Equation 3.12, we obtain

$$\begin{aligned}\text{dual}(\pi^{*\prime})[d] &= \begin{cases} \pi^*[a_0] & \text{if } d = a_k \\ \pi^*[b_0] & \text{if } d = b_\ell \\ \pi[d] & \text{otherwise} \end{cases} \\ &= \begin{cases} b_1 & \text{if } d = a_k \\ a_1 & \text{if } d = b_\ell \\ \pi[d] & \text{otherwise} \end{cases}\end{aligned}$$

Thus $\text{dual}(\pi^{*\prime})$ has a permutation cycle $(a_1\ a_2\ \cdots\ a_k\ b_1\ b_2\ \cdots\ b_\ell)$. This permutation cycle defines the vertex obtained by merging u and v and removing the edge e. All other vertices are unchanged. This shows that the underlying graph is that obtained by contracting e. □

In view of Lemma 3.4.8, we define *compression* of an edge e in an embedded graph G to be deletion of e in the dual G^*. We denote this operation by G/e. That is, $G/e = \text{dual}(G^* - e)$. Compression of an edge of an embedded graph yields an embedded graph.

In the case when e is not a self-loop, we refer to the operation as *contraction* of e.

What about the case of compression when e is a self-loop? We discuss this later when we study planar graphs.