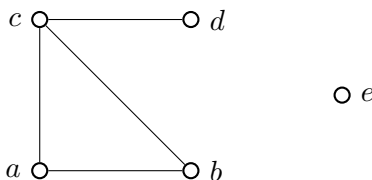


## 1 Paths and cycles

A *path* in a graph is a sequence of distinct vertices  $v_0, \dots, v_\ell$  such that  $\{v_i, v_{i+1}\}$  is an edge in the graph for all  $0 \leq i < \ell$ . We can also describe a path as a sequence of the  $k - 1$  edges  $\{v_0, v_1\}, \dots, \{v_{\ell-1}, v_\ell\}$ . The vertices  $v_0$  and  $v_\ell$  are the *endpoints* of the path; we also say that  $v_0$  is *connected to*  $v_\ell$ . The number  $\ell$  is the *length* of the path. (The definition allows paths of length zero.)

For example, in the graph given by the following diagram,  $a, c, d$  is a path of length 2 and  $a, b, c, d$  is a path of length 3. The sequence  $a, b, e$  is not a path because  $\{b, e\}$  is not an edge, and  $a, b, c, a$  is not a path because vertex  $a$  occurs more than once.



A *cycle* in a graph is a cyclic sequence<sup>1</sup> of distinct vertices  $v_1, \dots, v_\ell$  such that  $\{v_1, v_2\}, \dots, \{v_{\ell-1}, v_\ell\}, \{v_\ell, v_1\}$  are edges and they are all distinct. The number  $\ell$  is called the *length* of the cycle. For example, the above graph has only one cycle – the cycle  $a, b, c$  – and this cycle has length 3. (The definition does not allow cycles of length zero.)

If we do not make the requirement that  $v_1, \dots, v_\ell$  are all distinct, then we call  $v_1, \dots, v_\ell$  a *closed walk*. Any graph with at least one edge has an infinite number of closed walks: If  $\{u, v\}$  is an edge, then  $u, v$ ;  $u, v, u, v$ ;  $u, v, u, v, u, v$ , and so on are all closed walks.

## 2 Connectivity, trees, and forests

A graph is *connected* if for every pair of vertices  $u, v$ ,  $u$  is connected to  $v$ . The above graph is not connected because there exists no path from  $a$  to  $e$ .

A *connected component* of a graph  $G$  is a subgraph of  $G$  consisting of all vertices that are connected to a given vertex and all edges incident to them. This graph has the following two connected

---

<sup>1</sup>By “cyclic”, we mean that the sequences (1)  $v_1, v_2, \dots, v_\ell$ ; (2)  $v_2, \dots, v_\ell, v_1$ ; and (3)  $v_\ell, v_{\ell-1}, \dots, v_1$  all represent the same cycle.

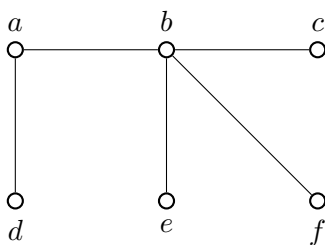
components:

$$(V_1, E_1), \quad \text{where } V_1 = \{a, b, c, d\} \text{ and } E_1 = \{\{a, b\}, \{a, c\}, \{b, c\}, \{c, d\}\}$$

$$(V_2, E_2), \quad \text{where } V_2 = \{e\} \text{ and } E_2 = \emptyset.$$

It should be clear that there can be no edges between vertices that belong to different connected components of the same graph.

A *forest* is a graph that has no cycles. A *tree* is a connected forest. A *leaf* in a forest is a node of degree 1. Here is a diagram of a tree with four leaves:



**Theorem 1.** *Every forest with  $n$  vertices and  $m$  edges has  $n - m$  connected components.*

*Proof.* We prove the theorem by induction on  $m$ .

**Base case  $m = 0$ :** If a forest has no edges, every vertex is its own connected component, so there are exactly  $n$  of them.

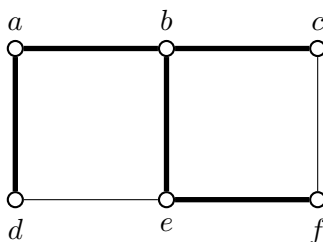
**Inductive step:** Assume that every forest with  $m$  edges has  $n - m$  connected components, where  $n$  is the number of vertices. Let  $G$  be a forest with  $n$  vertices and  $m + 1$  edges. Remove an arbitrary edge  $e$  from  $G$  (but leave its vertices alone). The resulting graph, call it  $G'$ , is a forest with  $n$  vertices and  $m$  edges, so by our inductive assumption it has  $n - m$  connected components.

Both vertices of  $e$  cannot belong to the same connected component of  $G'$ , for then the path between the endpoints of  $e$  together with  $e$  would form a cycle in  $G$ .

Therefore the vertices of  $G'$  are in two different connected components of  $G'$ . All other connected components of  $G'$  except for these two stay the same in  $G$ , so  $G$  has one fewer connected component than  $G'$ . Therefore  $G$  has exactly  $(n - m) - 1 = n - (m + 1)$  connected components.  $\square$

**Corollary 2.** *In a tree, the number of vertices equals the number of edges plus one.*

A *spanning tree* of a connected graph  $G$  is a subgraph of  $G$  that is a tree. Here is a diagram of a graph and one of its spanning trees. The edges of the spanning tree are marked by thick lines:



**Theorem 3.** *Every connected graph has a spanning tree.*

*Proof.* The proof is by induction on the number of edges. If the graph has no edges, then for it to be connected it must consist of a single vertex. This vertex is then a spanning tree.

Now assume that every connected graph with  $m$  edges has a spanning tree. Let  $G$  be a graph with  $m + 1$  edges. We consider two cases. If  $G$  is a tree, then  $G$  is its own spanning tree. Otherwise,  $G$  has a cycle. Take any edge  $e$  on this cycle and remove it. The remaining graph  $G'$  is connected: If any path between two vertices of  $G$  uses the edge  $e$ , it can be rerouted along the other edges of the cycle. Since  $G'$  has  $m$  edges, by the inductive hypothesis it has a spanning tree. Therefore  $G$  also has a spanning tree.  $\square$

The proof of this theorem tells us how to find a spanning tree in a connected graph: If the graph is a tree already, we are done. Otherwise, it must contain a cycle. Remove one of the edges in this cycle and recursively find a spanning tree in this subgraph. There are better algorithms for finding spanning trees that you will learn about in CSCI 3160.

Here is a corollary of Theorem 3 and Theorem 1:

**Corollary 4.** *Every graph with  $n$  vertices and  $m$  edges has at least  $n - m$  connected components.*

*Proof.* Let  $G$  be a graph with  $n$  vertices and  $m$  edges. By Theorem 3, every connected component of  $G$  has a spanning tree. Let  $F$  be the union of all these spanning trees. Then  $F$  is a forest with  $n$  vertices,  $m' \leq m$  edges, and the same number of connected components as  $G$ . By Theorem 1,  $F$  has  $n - m'$  connected components, so  $G$  also has  $n - m'$  connected components. Since  $m' \leq m$ , the number of connected components of  $G$  is at least as large as  $n - m$ .  $\square$

### 3 Bipartite graphs revisited

Recall that a graph is bipartite if its vertices can be partitioned into two sets  $L, R$  so that all edges have one vertex in  $L$  and one vertex in  $R$ .

**Theorem 5.** *A graph is bipartite if and only if it has no cycle of odd length.*

Before we prove this theorem, let us look at two familiar examples.



The first graph is not bipartite and the cycle  $a, b, c$  has odd length. The “reason” this graph is not bipartite is precisely the existence of this cycle: If we put vertex  $a$  on one side of the partition,  $b$  goes on the other side and  $c$  is on the same side, so the edge  $\{a, c\}$  is inconsistent with the partition. We will generalize this argument to rule out the possibility that a graph with a cycle of odd length can be bipartite.

In contrast, the second graph has no cycles of odd length and it is bipartite. One way to find the partition  $L, R$  is to start at an arbitrary vertex — say  $a$  — and put it in one set of the partition — say  $L$ . Then its neighbours  $b$  and  $d$  must be in  $R$ , their neighbours  $c$  and  $e$  must be in  $L$ , and their neighbour  $f$  must be in  $R$  again. We obtain the partition  $L = \{a, c, e\}$ ,  $R = \{b, d, f\}$ . In the proof, we will argue that as long as there are no odd length cycles, it is always possible to obtain a valid partition in a similar manner.

The following simple lemma will be useful to have:

**Lemma 6.** *If a graph has a closed walk of odd length, then it has a cycle of odd length.*

*Proof.* We prove the lemma by strong induction of then length  $\ell$  of the closed walk.

**Base case  $\ell = 1$ :** There are no closed walks of length one, so the base case holds trivially.

**Inductive step:** Assume that if a graph has a closed walk of odd length up to  $\ell$ , then it has an odd length cycle. Let  $v_1, \dots, v_{\ell+1}$  be a closed walk of length  $\ell + 1$  and assume  $\ell + 1$  is odd. If all the vertices in the closed walk are distinct, then  $v_1, \dots, v_{\ell+1}$  is a cycle of odd length. Otherwise, two of them must be the same, say  $v_i$  and  $v_j$  where  $1 \leq i < j \leq \ell + 1$ . Consider the closed walks:

$$v_i, v_{i+1}, \dots, v_{j-1} \quad \text{and} \quad v_j, \dots, v_{\ell+1}, v_1, \dots, v_{i-1}$$

The sum of the lengths of these two closed walks is  $\ell + 1$ , so each one has length strictly less than  $\ell + 1$ . Moreover,  $\ell + 1$  is an odd number, so one of them must have odd length. By the inductive assumption, the existence of this walk implies the existence of an odd cycle.  $\square$

*Proof of Theorem 5.* Assume the graph is bipartite with its vertices partitioned into  $L$  and  $R$ . Suppose  $v_1, \dots, v_\ell$  is a cycle for some  $\ell \geq 2$ . We will show that  $\ell$  is even. Without loss of generality, we may assume  $v_1$  is in  $L$ ; if not, we look at the sequence  $v_2, \dots, v_\ell, v_1$ , which describes the same cycle. Since  $v_1$  is in  $L$ ,  $v_2$  must be in  $R$ ,  $v_3$  must be in  $L$  again, and so on; so  $v_\ell$  is in  $L$  if  $\ell$  is odd and in  $R$  if  $\ell$  is even. Since  $\{v_\ell, v_1\}$  is an edge,  $v_\ell$  must be in  $R$ , so  $\ell$  is even.<sup>2</sup>

<sup>2</sup>If you want a very formal proof, you prove the proposition “For all  $1 \leq i \leq \ell$ ,  $v_i \in L$  if and only if  $i$  is even” by induction on  $i$ .

Now assume the graph, which we call  $G$ , has no cycles of odd length. We will prove that it is bipartite.

Suppose that we can show every *connected* graph that has no cycles of odd length is bipartite. Now take an arbitrary, not necessarily connected, graph  $G$  with no cycles of odd length. Then each connected component  $C$  of  $G$  has no cycles of odd length, so its vertices can be partitioned into two sets  $L_C$  and  $R_C$  so that there are no edges of  $C$  within either of  $L_C$  and  $R_C$ . Let  $L$  be the union of all such  $L_C$  and  $R$  be the union of all such  $R_C$ . Then  $L, R$  is a partition of the vertices of  $G$ . There are no edges within  $L$  and no edges within  $R$ , so  $G$  is bipartite.

Therefore, we can now assume *without loss of generality* that the graph  $G$  is connected in addition to having no cycles of odd length. We will show that  $G$  is bipartite. Let  $v_0$  be an arbitrary vertex and  $L$  and  $R$  be the following sets:

$$\begin{aligned} L &= \{w: \text{there exists a path of even length with endpoints } v_0, w\} \\ R &= \{w: \text{there exists a path of odd length with endpoints } v_0, w\} \end{aligned}$$

Since  $G$  is connected, every vertex must be connected to  $v_0$ , so it belongs to one of  $L$  or  $R$ . We will now show that  $L \cap R = \emptyset$ . The proof is by contradiction. Suppose there was a vertex  $u \in L \cap R$ . Since  $u$  is in  $L$ , there is an even length path from  $v_0$  to  $u$ :

$$v_0, v_1, \dots, v_s = u, \quad s \text{ is even}$$

Since  $u$  is in  $R$ , there is an odd length path from  $v$  to  $v_0$ . The reverse path is an odd length path from  $u$  to  $v$ :

$$u = v_s, v_{s+1}, \dots, v_{s+t} = v_0, \quad t \text{ is odd.}$$

Let us look at the sequence  $v_0, v_1, \dots, v_{s+t-1}$ . This is a closed walk of length  $s+t$ , which is an odd number. By Lemma 6,  $G$  has a cycle of odd length, which is a contradiction.

Therefore  $L, R$  is a partition of the vertices of  $G$ . It remains to show that there can be no edges within  $L$  and no edges within  $R$ . We first show there are no edges within  $L$ . For contradiction, suppose that there was an edge  $\{u, w\}$  such that  $u \in L$  and  $w \in L$ . By the definition of  $L$ , there must exist even length paths from  $w$  to  $v_0$  and from  $u$  to  $v_0$ . If we take the first path, then the second path in reverse, and finally follow the edge  $u, w$ , we obtain a closed walk of even + even + 1 = odd length. By Lemma 6, this contradicts our assumption. We now show there are no edges within  $R$ . The proof is analogous; the only difference is that the two paths now have odd length. The length of the resulting closed walk is now odd + odd + 1, which is still an odd number.  $\square$

We state a corollary of this theorem that will be useful for us shortly.

**Corollary 7.** *Let  $G$  be a graph whose edges are the union of two matchings. Then  $G$  is bipartite.*

*Proof.* We show that  $G$  has no cycles of odd length. By Theorem 5,  $G$  must be bipartite.

For contradiction, assume that there exists an odd length cycle  $v_1, v_2, \dots, v_\ell$  in  $G$ . Then the edge  $\{v_1, v_2\}$  must belong to at least one of the two matchings whose union forms the edges of  $G$ . Call this matching  $\Xi_1$ . The edge  $\{v_2, v_3\}$  cannot belong to  $\Xi_1$  for otherwise  $\Xi_1$  would not be a matching,

so it must belong to  $\Xi_2$ . Continuing the reasoning in this way, we conclude that for every  $1 \leq i < \ell$ ,  $\{v_i, v_{i+1}\}$  must be in  $\Xi_1$  if  $i$  is odd and in  $\Xi_2$  if  $i$  is even, so the edge  $\{v_{\ell-1}, v_\ell\}$  must be in  $\Xi_2$ . But then the edge  $\{v_\ell, v_1\}$  is in  $\Xi_1$ , and so  $v_1$  has two neighbours in  $\Xi_1$  ( $v_2$  and  $v_\ell$ ). This contradicts our assumption that  $\Xi_1$  is a matching.  $\square$

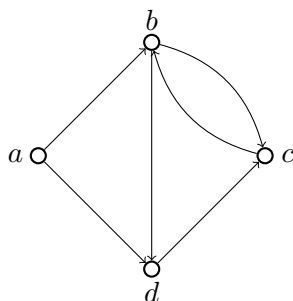
## 4 Directed graphs

A *simple directed graph* (or *digraph*)  $G$  consists of a nonempty set of vertices  $V$  and a set of directed edges  $E$ , each of which is an ordered pair of vertices  $(u, v)$  with  $u \neq v$ .

In a digraph,  $(u, v)$  and  $(v, u)$  represent different edges. One of them could be present in the graph, or both, or neither. In a diagram, we represent them as arrows from  $u$  to  $v$  and from  $v$  to  $u$ , respectively.

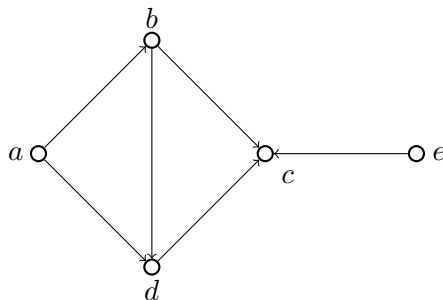
Here is a diagram of the digraph  $(V, E)$  where

$$V = \{a, b, c, d\} \quad \text{and} \quad E = \{(a, b), (a, d), (b, c), (b, d), (c, b), (d, c)\} :$$



The definitions of “path”, “closed walk”, and “cycle” in a digraph are identical to those for graphs. For example,  $a, b, d, c$  is a path of length 4 in  $G$  and  $b, d, c$  is a cycle of length 3.  $G$  has also a cycle of length 2, namely  $b, c$ . In contrast, a graph cannot have a cycle of length 2.

A *topological sort* of a digraph  $G$  is an ordering of all its vertices in a sequence so that for every edge  $(u, v)$ , vertex  $u$  comes before vertex  $v$  in the sequence. The above digraph does not have a topological sort. The vertices  $b$  and  $c$  cannot be ordered because both  $b$  must come before  $c$  and  $c$  must come before  $b$  in the sequence. This is impossible. In contrast, one topological sort of the digraph in the following diagram is the sequence  $a, b, d, e, c$ :



**Theorem 8.** *A digraph has a topological sort if and only if it has no cycles.*

We will need the following lemma. A *sink* is a vertex with no outgoing edge. Similarly, a *source* is a vertex with no incoming edge, and a vertex is *internal* if it is neither a source nor a sink.

**Lemma 9.** *A digraph with no cycles has a sink.*

*Proof.* We prove the contrapositive: A digraph with no sink has a cycle. Assume that  $G$  has no sink. Then every vertex in  $G$  has at least one outgoing edge. Visit vertices according to the following procedure, starting from an arbitrary vertex  $v_0$ : Keep moving to an adjacent vertex via an outgoing edge. When you reach the first previously visited vertex, stop. Then the sequence of visited vertices will have the following form:

$$v_0, v_1, \dots, v_{k-1}, v_k$$

where  $v_0, \dots, v_{k-1}$  are all distinct and  $v_k = v_i$  for some  $i$  between 0 and  $k-1$ . Then  $v_i, \dots, v_{k-1}$  is a cycle in  $G$ , so  $G$  has a cycle.  $\square$

*Proof of Theorem 8.* First we show that if a digraph has a topological sort, it cannot have a cycle. For contradiction, suppose  $v_1, \dots, v_\ell$  is a cycle. Then  $v_2$  must come after  $v_1$  in the sequence,  $v_3$  must come after  $v_2$ , and so on until  $v_\ell$ . So  $v_\ell$  must come after  $v_1$ . Since  $(v_\ell, v_1)$  is an edge,  $v_1$  must come after  $v_\ell$ . Contradiction.

Now we show that if a digraph  $G$  has no cycles, then it must have a topological sort. We prove this by induction on the number of vertices  $n$ .

**Base case  $n = 1$ :** If  $G$  has one vertex  $v$ , then the sequence  $v$  is a topological sort of  $G$ .

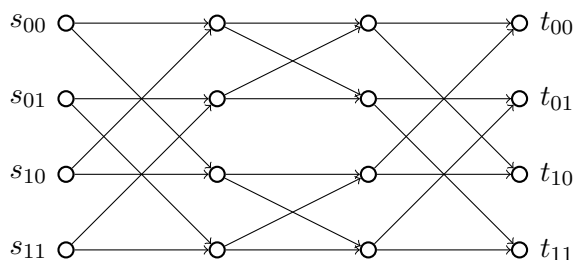
**Inductive step:** Assume every digraph with no cycles on  $n$  vertices has a topological sort. Let  $G$  be a digraph with no cycles on  $n + 1$  vertices. By Lemma 9,  $G$  has a sink  $t$ . Remove  $t$  and all incoming edges from  $G$ . By the inductive assumption, the remaining graph has a topological sort sequence  $v_1, \dots, v_n$ . Then  $v_1, \dots, v_n, t$  is a topological sort of  $G$ .  $\square$

A digraph with no cycles is called an *acyclic digraph* or a DAG (directed acyclic graph).

## 5 Switching networks

A collection of paths is *vertex disjoint* if all vertices in all paths in the collection are distinct. A *switching network* for  $N$  packets is an acyclic digraph with  $N$  sources and  $N$  sinks (and possibly some internal vertices) so that for every possible pairing of sources and sinks there exists a vertex-disjoint collection of paths from each source to the corresponding sink.

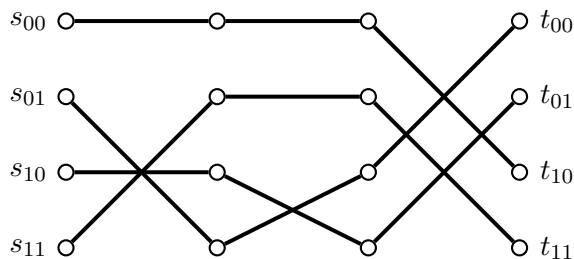
Here is a diagram of a switching network, which we'll call  $B_2$ , for 4 packets:



For example, suppose we pair up the sources and sinks like this:

$$\{s_{00}, t_{10}\}, \{s_{01}, t_{00}\}, \{s_{10}, t_{01}\}, \{s_{11}, t_{11}\}$$

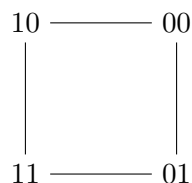
The following diagram shows the existence of vertex-disjoint paths from every source to the corresponding sink:



Here is how I found these paths. The subgraph connecting the middle nodes splits into a “top component” and a “bottom component”. Among the packets 00 and 10, one must travel through the top component and the other one through the bottom component, for otherwise they would clash at the first intermediate vertex. The same conclusion holds for the packets 01 and 11. The two packets that are supposed to arrive at  $t_{00}$  and  $t_{10}$  — these are the packets 01 and 00 — must also travel through different middle components. Similarly, the packets 10 and 11, which are supposed to reach  $t_{01}$  and  $t_{11}$ , should be routed through different middle components.

We can represent this information in a *constraint graph* whose vertices are packets and whose edges connect pairs that cannot be routed through the same middle component:

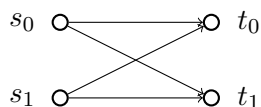




The constraint graph is bipartite: The vertices can be partitioned into the sets  $T = \{00, 11\}$  and  $B = \{10, 01\}$ . We route the packets in  $T$  through the top component and the packets in  $B$  through the bottom component. This uniquely specifies the first and last edge in each path (as in the above example). The middle edges are then easy to figure out.

How can we know for sure that  $B_2$  is a switching network? We need to verify that all possible source-sink pairings admit edge-disjoint paths. It turns out that  $B_2$  has a special, recursive structure that guarantees this feature. To make this structure more apparent, we will show how to construct, for every  $n \geq 1$ , a switching network  $B_n$  with  $2^n$  sources and  $2^n$  sinks, provided we have already constructed  $B_{n-1}$ . The digraph  $B_n$  is called the Beneš network (pronounced *Benes*) after its inventor.

The “base” digraph  $B_1$  looks like this:



Now we show how to construct the digraph  $B_{n+1}$ , assuming we have already constructed  $B_n$ . Recall that  $B_n$  has  $2^n$  sources and  $2^n$  sinks:

1. Take *two* disjoint (no shared vertices or edges) copies of  $B_n$ , which we will call the top copy and the bottom copy.
2. Label the sources of the top copy of  $B_n$  by the symbols  $u_{0x}$ , where  $x$  ranges over all  $\{0, 1\}$  strings of length  $n$ . Label the sources of the bottom copy of  $B_n$  by the symbols  $u_{1x}$ , where  $x$  ranges over all  $\{0, 1\}$  strings of length  $n$ . Label the sinks of the top and bottom copies of  $B_n$  by the symbols  $v_{0x}$  and  $v_{1x}$ , respectively, where  $x$  ranges over all  $\{0, 1\}$  strings of length  $n$ .
3. Create  $2^{n+1}$  sources  $s_y$  and  $2^{n+1}$  sinks  $t_y$  for  $B_{n+1}$ , where  $y$  ranges over all  $\{0, 1\}$  strings of length  $n + 1$ .
4. Add the edges  $(s_{ax}, u_{bx})$  and  $(v_{bx}, t_{ax})$  for every bit  $a \in \{0, 1\}$ , every bit  $b \in \{0, 1\}$ , and every  $\{0, 1\}$  string  $x$  of length  $n$ .

You can check that our digraph  $B_2$  was indeed constructed from  $B_1$  according to this specification.

**Theorem 10.** *For every  $n \geq 1$ , the digraph  $B_n$  is a switching network.*

*Proof.* We prove the theorem by induction on  $n$ . In the base case  $n = 1$ , there are two possibilities: The source-sink matching is either  $\{s_0, t_0\}, \{s_1, t_1\}$  or  $\{s_0, t_1\}, \{s_1, t_0\}$ . In either case, there are vertex disjoint paths from each source to the corresponding sink.

For the inductive step, we assume that  $B_n$  is a switching network. We will show that  $B_{n+1}$  is also a switching network. Let  $\Pi$  be a pairing of the  $2^{n+1}$  sources  $s_y$  and the  $2^{n+1}$  sinks  $t_y$  in  $B_{n+1}$ . We need to show that all pairs in  $\Pi$  can be connected by edge-disjoint paths.

We define an *undirected* “constraint” graph  $G$  as follows. The vertices of  $G$  are all the  $2^{n+1}$   $\{0, 1\}$  strings of length  $n + 1$ . The edges of  $G$  are the union of two perfect matchings. The first matching consists of all the edges  $\{0x, 1x\}$ , where  $x$  ranges over all  $\{0, 1\}$  strings of length  $n$ . The second matching consists of all the edges  $\{y, y'\}$  such that  $s_y$ 's and  $s_{y'}$ 's partners in  $\Pi$  differ only in their first coordinate (that is, they are  $t_{0x}$  and  $t_{1x}$  for some  $x$ ). You can easily verify that this is a matching.

By Corollary 7, the graph  $G$  is bipartite. Let  $T, B$  be a partition of its vertices so there are no edges within  $T$  or within  $B$ . We now describe the edge-disjoint paths from sources to sinks of  $\Pi$ . We will specify each path by describing its first edge, its last edge, and its middle edges. We will do the middle edges later:

1. **First edge:** For every vertex  $s_{bx}$  (where  $b$  is a bit and  $x$  is a string of length  $n$ ), take the edge from  $s_{bx}$  to  $u_{0x}$  if  $bx \in T$ ; take the edge from  $s_{bx}$  to  $u_{1x}$  if  $bx \in B$ .
2. **Last edge:** For every vertex  $t_{bx}$ , Let  $s_y$  be  $t_{bx}$ 's partner vertex in  $\Pi$ . Take the edge from  $v_{0x}$  to  $t_{bx}$  if  $y \in T$ ; take the edge from  $v_{1x}$  to  $t_{bx}$  if  $y \in B$ .

We will now show three properties: (1) the set of first edges  $\Phi_1$  is a matching; (2) the set of last edges  $\Phi_2$  is a matching; and (3) For every pair  $\{s_y, t_{y'}\}$  in  $\Pi$ ,  $s_y$ 's match in  $\Phi_1$  and  $t_{y'}$ 's match in  $\Phi_2$  both belong to the same copy of  $B_n$ .

Once we have these properties, our inductive assumption about  $B_n$  guarantees the existence of edge-disjoint paths from  $s_y$ 's match in  $\Phi_1$  to  $t_{y'}$ 's match in  $\Phi_2$  for every pair  $\{s_y, t_{y'}\}$  in  $\Pi$ . These paths form the middle edges from the sources to the sinks of  $\Pi$ . Together with the first and last edges, they describe a collection of edge-disjoint paths between all source-sink pairs of  $\Pi$ , the existence of which is what we wanted to prove.

So it remains to show properties (1), (2), and (3). We show (1) first. For contradiction, assume  $\Phi_1$  is not a matching. Then two of the first edges share the same vertex  $u_{bx}$ . The only two edges that share this vertex are  $(s_{0x}, u_{bx})$  and  $(s_{1x}, u_{bx})$ . This is not possible because  $\{0x, 1x\}$  is an edge in (the first matching of)  $G$ , and so one of the two must be in  $T$  and the other one in  $B$ . So one of the edges must point to  $u_{0x}$  and the other one must point to  $u_{1x}$ , a contradiction.

The proof of (2) is similar: If  $\Phi_2$  is not a matching, two of the last edges must share the same vertex  $v_{bx}$ . These are the edges  $(v_{bx}, t_{0x})$  and  $(v_{bx}, t_{1x})$ . So the partners of  $t_{0x}$  and  $t_{1x}$  in  $\Pi$ , which form an edge in (the second matching of)  $G$ , must both be in the same set of the partition  $T, B$ , a contradiction.

Property (3) is immediate: If  $s_y$  is paired to  $t_{y'}$  in  $\Pi$ , then both  $s_y$ 's match in  $\Phi_1$  and  $t_{y'}$ 's match in  $\Phi_2$  are in the top copy of  $B_n$  if  $y$  is in  $T$  and the bottom one if  $y$  is in  $B$ . In either case, they

are in the same copy of  $B_n$ . □

The vertex-disjoint paths in our example were obtained by following the procedure described in the proof. To check your understanding, it is a good idea to walk through the steps of the proof keeping this example in mind.

## References

This lecture is based on Chapters 5 and 6 and Section 7.7 of the text *Mathematics for Computer Science* by E. Lehman, T. Leighton, and A. Meyer.

The textbook has a detailed discussion about the advantages and disadvantages of various types of switching networks. There is yet another type of network, called a *superconcentrator*, that (according to the measures in the book) outperforms the Beneš network when the number of packets is large. If you are interested, you can read the fascinating story of superconcentrators in Section 1.1 of the survey *Expander graphs and their applications* by Hoory, Linial, and Wigderson.