CS 740, Fall 1995 Homework Assignment 5

Sample Answers

1. Bisection of Hypercube (40%)

Let N = 2^n.

Upper bound: First, we label the nodes of the hypercube H_N with binary number so that an edge connects two nodes iff the binary number of the two nodes differ in exactly one bit. Then, we remove the edges in which the binary number of two end point is only different in the first bit. Given a number with binary representation $1x_2x_3...x_n$, there are exactly one number ($0x_2x_3...x_n$) whose binary representation differ from this in only the first bit, therefore there are exactly N/2 such edges. Now, these edges bisect the hypercube H_N into two sub-hypercubes with N/2 nodes each. There is no edge between these two sub-hypercubes, because such edge would involve changing the value of the first bit, and we remove all such edges. Hence the bisection width of H_N is at most N/2 edges.

Lower bound: Consider an embedding of the complete DIRECTED graph K_N on N nodes into H_N in the following manner. Map each node into the node with corresponding number in H_N. The edge between two distinct nodes x and y with binary representation $x_1x_2...x_n$ and $y_1y_2...y_n$ in K_N is mapped into a path from x to y in H_N by flipping each bit i, from 1 to n, for which $x_i \neq y_i$. Note that the path from x to y may not be the same as the path from y to x. This is the reason to use DIRECTED graph.

In K_N, these N nodes can be divided into 2 disjoint set of N/2 nodes. The number of directed edges from one set to the other is (N/2)*(N/2)*2=N^2/2. Hence, the bisection width of K_N is at least N^2/2. Now, let us consider the number of paths crossing an edge between $z_1...z_{i-1}z_iz_{i+1}...z_n$ and $z_1...z_{i-1}\bar{z}_iz_{i+1}...z_n$. A path from x to y (y to x) crosses this edge iff y_j (x_j)=z_j for $j<i$, y_i (x_i)=$\bar{z}_i$, and x_j (y_j)=z_j for $j>=i$. There are 2^n pairs of nodes x and y in K_N who have paths crossing the edge. Since N=2^n, there are N paths crossing the edge.

Assume that we have a bisection of the hypercube which size is less than N/2. Then the number of paths crossing these bisection edges is less than N^2/2. This contradicts the bisection width of K_N shown above. Hence the bisection width of the hypercube is at least N/2. Therefore, the bisection width is exactly N/2.

Extra Credit(10%): Assume n is even. Let the bisect nodes B are the nodes which have n/2 ones and n/2 zeros in their binary representation. Let A be the set of nodes which has less than n/2 ones and C be the set of nodes which has more than n/2 ones. These two disjoint sets have the same size by symmetry. For any node a in A and c in C, the distance between a and c is greater than 2, therefore there is no edge between a and c. Hence B bisect the hypercube into sets A and C. Now, the number of nodes in B is $\binom{n}{n/2} = n!/\left(\frac{n}{2}!\right)^2$. Using Stirling approximation, the size of B is $O(N/\sqrt{\log N})$. When n is odd, let B be the set of nodes which have n/2 ones or n/2+1 ones. This bisector still works, but is only a constant factor larger.

2. The Omega Network(30%)

In problem, you have to prove that the butterfly network can be mapped into Omega network, and vice-versa. If you just did one part, you got 15 credits. If you just show the node mapping functions, you got 10 points.

Let N (2^n) be the number of node at each layer in the butterfly or Omega networks. There are n+1 layers in both networks. In the butterfly network, any given node <x,k>, where x (the position of node within the layer) is in [0,N-1] and k (the layer) is in [0,n-1], is connected to two nodes B0(x,k) = <clear(x,n-k-1),k+1> and B1(x,k) = <set(x,n-k-1),k+1> in the k+1 layer where clear(x,n-k-1) clears the n-k-1 bit in x and set(x,n-k-1) sets the n-k-1 bit in the x. Similarly, any given node <x,k> in the omega network is connected to two nodes O0(x,k)=<2i mod N, k+1> and O1(x,k)=<2i+1 mod N, k+1>, where 2i mod N is equivalent to clear(x<-1,0) and 2i+1 mod N is equivalent to set(x<-1,0). x<-k means left rotating x by k bits and x->k means right rotating x by k bits. Notice that the Omega network has the property that the node connections between two layers are independent of the layer.

Notice that node <x,k> of the Omega network is the same as node OtoB(x,k)=<x->k,k> in the Butterfly network. Similarly, node <x,k> in the Butterfly network is the same as node BtoO(x,k) = <x<-k,k> in the Omega network. Figure 1 shows an 8-input Butterfly network and Figure 2 shows it corresponding Omega network after the mapping. In figure 2, the nodes are labelled as the original values of the Butterfly network. Obviously, BtoO is the inversion function of OtoB and vice-versa (i.e. BtoO(OtoB(x,k)) = <x,k> and OtoB(BtoO(x,k)) = <x,k>), and both BtoO and OtoB are bijective.

To show that these two networks are isomorphic, we show that the edge in the butterfly network can be mapped to the omega network and vice-versa. We only need to show that the following equations hold for any layer k between 0 and n-1: OtoB(O0(BtoO(x,k))) = B0(x,k), OtoB(O1(BtoO(x,k))) = B1(x,k), BtoO(B0(OtoB(x,k))) = O0(x,k), and BtoO(B1(OtoB(x,k))) = O1(x,k).

OtoB(O0(BtoO(x,k))) = OtoB(O0(x<-k,k)) = OtoB(clear((x<-k)<-1,0),k+1) = OtoB(clear(x<-(k+1),0),k+1) = <(clear(x<-(k+1),0))->(k+1),k+1> = <clear(x,n-k-1),k+1> = B0(x,k).

OtoB(O1(BtoO(x,k))) = OtoB(O1(x<-k,k)) = OtoB(set((x<-k)<-1,0),k+1) = OtoB(set(x<-(k+1),0),k+1) = <(set(x<-(k+1),0))->(k+1),k+1> = <set(x,n-k-1),k+1> = B1(x,k).

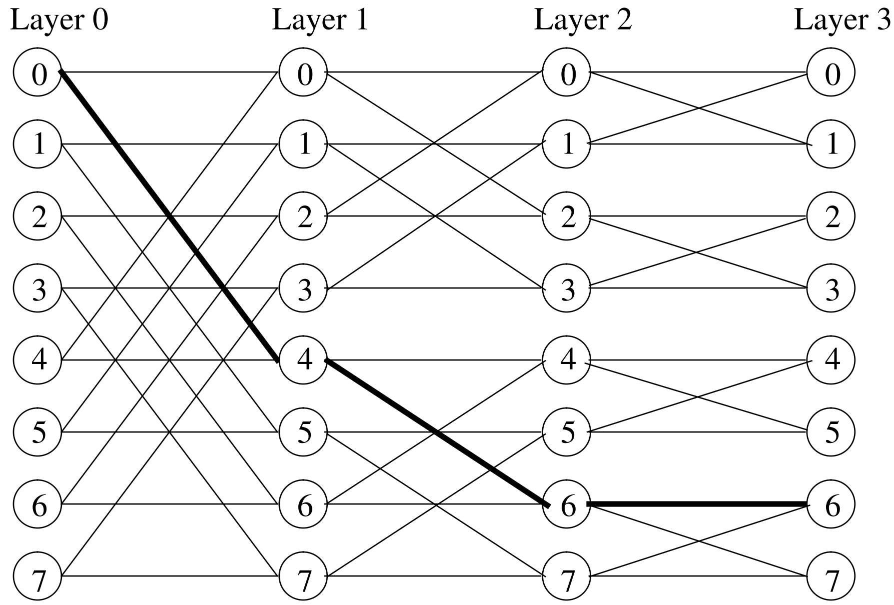

Figure 1. An 8-input butterfly network.

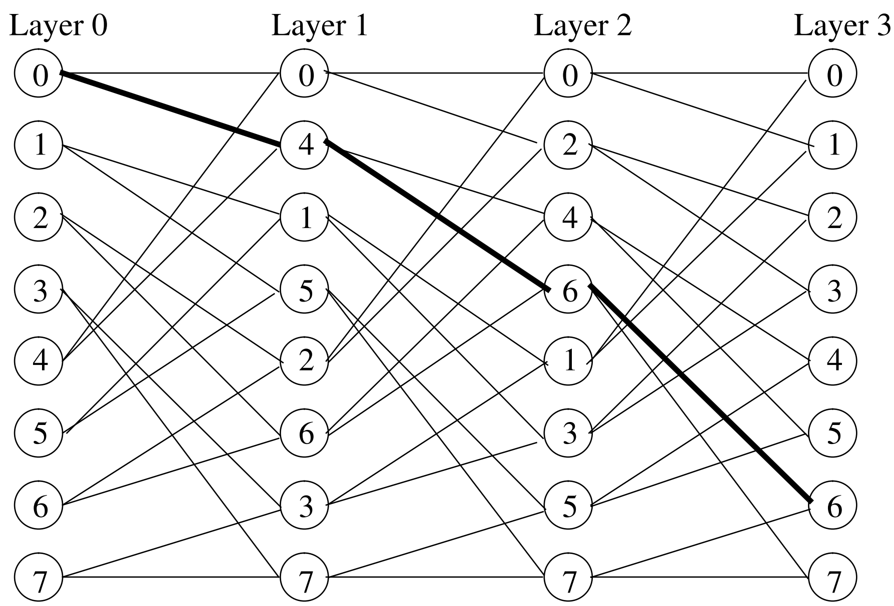

Figure 2. An 8-input Omega network after mapping from the butterfly network.

BtoO(B0(OtoB(x,k))) = BtoO(B0(x->k,k)) = BtoO(clear(x->k, n-k-1), k+1) = <(clear(x->k,n-k-1))<-(k+1),k+1>=<clear(x<-1,0),k+1> = O0(x,k).

BtoO(B1(OtoB(x,k))) = BtoO(B1(x->k,k)) = BtoO(set(x->k, n-k-1), k+1) = <(set(x->k,n-k-1))<-(k+1),k+1>=<set(x<-1,0),k+1> = O0(x,k).

These four equations show us that there is an edge between two nodes in butterfly network if and only if there is a corresponding edge in the omega network. Therefore, these two network is isomorphic.

3. Set difference(30%)

If you wrote a non-parallel version of set_diff function, it costs you 10 points

```
function set_diff(a, b) =
    if (#a == 0 or #b==0) then a
    else let pivot   = a[#a/2];
             lessera = {e in a| e < pivot};
             equala   = {e in a| e == pivot};
             greatera = {e in a| e > pivot};
             lesserb = {e in b| e < pivot};
             equalb = {e in b| e == pivot};
             greaterb = {e in b| e > pivot};
             result = {set_diff(u, v): u in [lessera,greatera]; v in [lesserb,greaterb]};
     in if (#equalb == 0) then result[0] ++ equala ++ result[1]
        else result[0] ++ result[1] $
```

4.Grade Distributions

above 90: 6

89-80: 11

79-70: 3