Efficient Algorithms
Summer term 2013

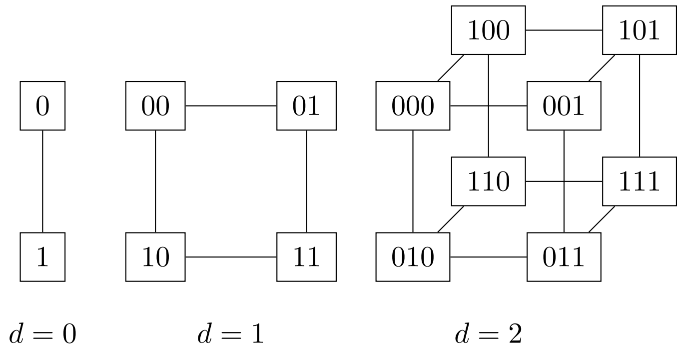
Institut für Theoretische Informatik
Leibniz Universität Hannover

Solution to exercise sheet 11

25.06.2013

Exercise 1: A *hypercube* of dimension d consists of 2^d processors which are numbered by the d-ary dual-representation of processor ids. Two processors are connected through a communication channel if their ids have a hamming distance of 1 (i.e., their dual-representation differs in exactly one bit). The structure of a hypercube is recursively defined:

- The hypercube of dimension $d+1$ choose two hypercubes H_1, H_2 of dimension d. The ids of H_1 are preceded with a 1 and the ids of H_2 with a 0. Then connect the corresponding vertices with an edge.



1. What are the attribute values (defined on the last exercise sheet) of this topology?
2. How many vertices and how many edges are in a hypercube of dimension d?
3. Given a hypercube of dimension d. What is the shortest path to pass a message from vertex u to the vertex v?
4. Construct an algorithm which computes the sum of the elements of a given array A of length $n = 2^d$ on a hypercube of dimension d running in $O(\log n)$ steps.
5. Construct an algorithm which distributes a value stored in p_0 through the complete net in $O(d)$ steps in a hypercube of dimension d.

Solution:

1. **Diameter** d

   Degree d

Bisection width 2^{d-1}

2. Vertices: 2^d, Edges: $d \cdot 2^{d-1}$.

3. The shortest path between two different vertices u, v is $1 \leq |u \oplus v|_1 \leq d$.

4. We assume that $A[i]$ is stored in the local variable B of processor i. The sum $\sum_{i=0}^{n} A[i]$ is stored at the end of the computation in P_0.

Algorithm 1: sum $\langle$Hypercube$\rangle$

N, p: integer; `//` `initialized` $0 \leq p < 2^d$
B, C: real; `//` B `contains` $A[p]$
m, d: integer;
$d := \log N$; `// Assumption:` $N = s^d = n$
for $m := d - 1$ *down to* 0 **do**
  if $2^m \leq p < 2^{m+1}$ **then** `send(`$B, p\&(2^m - 1)$`);` `// bitwise and`
  if $p < 2^m$ **then**
    `receive(`$C, p||2^m$`);` `// bitwise or`
    $B := B + C$;

- In the first `if`-condition the processors with the ids $(\overbrace{0\cdots01\underbrace{?\cdots?}_{m}}^{d})$ are chosen.

  They send to the processors $(\overbrace{0\cdots0\underbrace{?\cdots?}_{m}}^{d})$

- in the second `if`-condition the processors with the ids $(\overbrace{0\cdots00\underbrace{?\cdots?}_{m}}^{d})$ are chosen and they receive from the processors $(\overbrace{0\cdots01\underbrace{?\cdots?}_{m}}^{d})$

- The algorithm runs in $O(\log n)$ steps.

5. Here we state an algorithm which distributes a variable x store in p_0 through the complete net. At first, p_0 sends x to p_1, then both to p_2 and p_3, and so on and

Algorithm 2: distribute $\langle$Hypercube$\rangle$

N, p: integer; `// initialized,` $0 \le p < N = 2^d$
x: real;
d, i: integer;
$d := \log N$;
for $i = 0$ *to* $d - 1$ **do**
    if $p < 2^i$ **then** `//` $p = (0 \cdots 0\underbrace{? \cdots ?}_{i})$
        `send(`$x, p || 2^i$`);`
    if $2^i \le p < 2^{i+1}$ **then** `//` $p = (0 \cdots 01\underbrace{? \cdots ?}_{i})$
        `receive(`$x, p \& (2^i - 1)$`);`

so forth. The algorithm runs in $O(\log n)$.

□