# Suggested Solutions – Exercises of Lecture 8 – MVP'06

Alexandre David

### Exercise 5.1 - Amdahl's law

$$S = \frac{W}{T_P} = \frac{W}{W_S + \frac{W - W_S}{p}} \leq \frac{W}{W_S} \; \forall p > 0$$

### Exercise 5.2

On a single processor, 11 arcs are traversed by FDS before the solution is found. On two processors fours arcs are traversed by $P_1$ before it finds the solution. In Figure 5.3 (page 201), the cost was on counting the states explored, here we count the arcs. The speedup is $11/4 = 2.75$. The anomaly is due to the fact that in Figure 5.10(b) (page 228), the algorithm being executed is not the same as in Figure 5.10(a). The parallel algorithm performs less overall work than the serial algorithm. The algorithm in Figure 5.10(b) is not a true DFS. If a single processor alternates between the nodes of the left and right sub-trees (which emulates the parallel algorithm), then the serial algorithm traverses only 8 arcs and the speedup becomes 2.

### Exercise 5.3

As a reminder the degree of concurrency here corresponds the the maximal degree of concurrency in chapter 3. Table gives the answers. The overhead function is computed by using $E = \frac{W}{pT_P} \implies$

| Graph | (a) | (b) | (c) | (d) |
|---|---|---|---|---|
| Degree of concurrency (DC) | $2^{n-1}$ | $2^{n-1}$ | $n$ | $n$ |
| Longest path | $\log(n)$ | $\log(n)$ | $2n-1$ | $n$ |
| Number of processors | $2^n - 1$ | $2^n - 1$ | $n^2$ | $\frac{n(n+1)}{2}$ |
| Maximum possible speedup | $\frac{2^n-1}{\log(n)}$ | $\frac{2^n-1}{\log(n)}$ | $\frac{n^2}{2n-1}$ | $\frac{n+1}{2}$ |
| Longest path if $p = \text{DC}/2$ | $\log(n)+1$ | $\log(n)+1$ | $\frac{n}{2}+\frac{n}{2}+n+n-2$ | $\frac{n}{2}+\frac{n}{2}+\frac{n}{2}$ |
| Speedup for $p = \text{DC}/2$ | $\frac{2^n-1}{\log(n)+1}$ | $\frac{2^n-1}{\log(n)+1}$ | $\frac{n^2}{3n-2}$ | $\frac{n+1}{3}$ |
| Efficiency for $p = \text{DC}/2$ | $\frac{1}{\log(n)+1}$ | $\frac{1}{\log(n)+1}$ | $\frac{1}{3n-2}$ | $\frac{2}{3n}$ |
| Overhead ($T_0$) for $p = \text{DC}/2$ | $(2^n-1)\log(n)$ | $(2^n-1)\log(n)$ | $3n^2(n-1)$ | $\frac{n(n+1)(3n-2)}{4}$ |

Table 1: Characteristics of graphs of Figure 5.11 (page 229).

$pT_P = \frac{W}{E}$ in $T_0 = pT_p - W = W(\frac{1}{E} - 1)$.

### Exercise 5.9

We use the expression $T_P = n/p - 1 + 11\log(p)$. We have the constraint

$$512 \geq \frac{n}{p} - 1 + 11\log(p) \implies (512 - 11\log(p))p \geq n$$

that gives us the largest $n$ corresponding to a $p$ in Table . In general it is not possible to solve an arbitrarily large problem with a fixed amount of time provided an unlimited number of processors. For any parallel system with an isoefficiency function greater than $\Theta(p)$, a plot between $p$ and

| $p$ | 1 | 4 | 16 | 64 | 256 | 1024 | 4096 |
|---|---|---|---|---|---|---|---|
| $\log(p)$ | 0 | 2 | 4 | 6 | 8 | 10 | 12 |
| $n$ | 513 | 1964 | 7504 | 28608 | 108800 | 412672 | 1560576 |

Table 2: Largest $n$ for varying $p$.

the size of the largest problem that can be solved in a given time using $p$ processors will reach a maximum. It can be shown that for cost-optimal algorithms, the problem size can be increased linearly with the number of processors while maintaining a fixed execution time iff the isoefficiency fjnction is $\Theta(p)$.