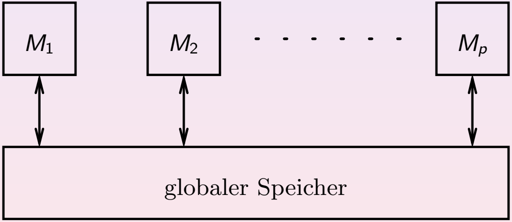
PRAMs

A PRAM (parallel random access machine) consists of p many identical processors $M_1, \ldots, M_p$ (RAMs).

- Processors can read from/write to a shared (global) memory.
- Processors work synchronously.



PRAMs

Different variants:

- CRCW (Concurrent Read Concurrent Write)
- CREW (Concurrent Read Exclusive Write)
- EREW (Exclusive Read Exclusive Write)
- ERCW (Exclusive Read Concurrent Write)

randomized PRAMs: Processors may toss coins.

The Class NC

Problems which are "efficiently parallelizable".

NC is called Nick's Class (after Nick Pippenger).

Definition: A problem belongs to the class NC, if it can be solved on a PRAM such that for an input of length n, we:

- use only n^d many processors (for a constant d), and
- spend time $(\log n)^c$ (for a constant c).

The class is robust to minor changes of the machine model: for instance, it doesn't matter, whether we use the CRCW/CREW/EREW/ERCW PRAM-model.

The question $\mathrm{NC} \stackrel{?}{=} \mathrm{P}$ is still open.

Luby's Algorithm

Recall that an independent set of an undirected graph $G = (V, E)$ is a subset $I \subseteq V$ such that $(u, v) \notin E$ for all $u, v \in I$.

Goal: For a given undirected graph $G = (V, E)$, find an independent set $I \subseteq V$ of G, which is maximal under inclusion, i.e., if $I \subseteq J$ for an independent set J then $I = J$.

We want to do this in NC, i.e., in polylogarithmic time using polynomially many processors.

Our first solution will be a randomized NC-algorithm.

For a set $U \subseteq V$ of nodes let
$N(U) = \{v \in V \mid \exists u \in U : (u, v) \in E\}$ be the set of neighbors of U.

Luby's Algorithm

Luby's algorithm works in rounds. In every round we calculate an independent set I in the current graph G and remove $I \cup N(I)$ (and all edges that are incident with a node from $I \cup N(I)$) from G.

We repeat this until the graph is empty. The calculated independent set is the union of the independent sets calculated in the rounds.

A single round, where $d(v) = |N(v)|$ for $v \in V$:

- In parallel: for every node $v \in V$, put v with probability $\frac{1}{2d(v)}$ into a set S (isolated nodes can be put into S into a preprocessing step), independently from the other nodes (i.e., $\Pr(\bigwedge_{i=1}^{k} v_i \in S) = \prod_{i=1}^{k} \Pr(v_i \in S)$).
- In parallel: For every $(u, v) \in E$ with $u, v \in S$, remove from S the node with the smaller degree (break ties arbitrarily). Call the remaining set I; it is an independent set.

Luby's Algorithm

- A single round can be done in constant time using $\mathcal{O}(|V|^2)$ processors.
- We will show that the *expected value* of the number of rounds is in $\mathcal{O}(\log |E|)$.
- First step: We show that the expected number of edges that are deleted in every round is at least $\frac{1}{72}$ of the total number of edges.

Luby's Algorithm

Lemma

For every node v: $Pr(v \in I) \geq \frac{1}{4d(v)}$

Proof: We will show that

$$\Pr(v \notin I \mid v \in S) \;\; \leq \;\; \frac{1}{2}$$

Then we obtain:

$$\begin{aligned}\Pr(v \in I) &= \Pr(v \in I \mid v \in S) \cdot \Pr(v \in S) \\ &\geq \frac{1}{2} \cdot \Pr(v \in S) = \frac{1}{4d(v)}.\end{aligned}$$

Luby's Algorithm

We have

$$\begin{aligned}
\Pr(v \notin I \mid v \in S) &\leq \Pr(\exists u \in L(v) : u \in S \mid v \in S) \\
\text{where } L(v) &= \{u \in N(v) \mid d(u) \geq d(v)\}\,.
\end{aligned}$$

Thus:

$$\begin{aligned}
\Pr(v \notin I \mid v \in S) &\leq \sum_{u \in L(v)} \Pr(u \in S \mid v \in S) \\
&= \sum_{u \in L(v)} \Pr(u \in S) \qquad \text{(independence)} \\
&= \sum_{u \in L(v)} \frac{1}{2d(u)} \\
&\leq \sum_{u \in L(v)} \frac{1}{2d(v)} \leq \frac{1}{2}, \text{ since } L(v) \subseteq N(v).
\end{aligned}$$

Luby's Algorithm

Definition: A node $v \in V$ is *good*, if

$$\sum_{u \in N(v)} \frac{1}{2d(u)} \geq \frac{1}{6}$$

(intuition: many neighbors with small degree), otherwise v is *bad*.
An edge $(u, v) \in E$ is *good*, if u or v is good, otherwise it is *bad*.

Lemma

For a good node $v \in V$ we have $Pr(v \in N(I)) \geq \frac{1}{36}$.

Proof:
Case 1: $\exists u \in N(v) : \frac{1}{2d(u)} > \frac{1}{6}$.
Then, by the previous lemma,

$$\Pr(v \in N(I)) \geq \Pr(u \in I) \geq \frac{1}{4d(u)} > \frac{1}{12} > \frac{1}{36}.$$

Luby's Algorithm

Case 2: $\forall u \in N(v) : \frac{1}{2d(u)} \leq \frac{1}{6}$.
Then there exists $M(v) \subseteq N(v)$ with $\frac{1}{6} \leq \sum_{u \in M(v)} \frac{1}{2d(u)} \leq \frac{1}{3}$.
Thus

$$
\begin{aligned}
\Pr(v \in N(I)) &\geq \Pr(\exists u \in M(v) : u \in I) \\
&\geq \sum_{u \in M(v)} \Pr(u \in I) - \sum_{u,w \in M(v), u \neq w} \Pr(u \in I \wedge w \in I) \\
&\geq \sum_{u \in M(v)} \frac{1}{4d(u)} - \sum_{u,w \in M(v), u \neq w} \Pr(u \in S \wedge w \in S) \\
&\quad \text{(independence)} \\
&\geq \sum_{u \in M(v)} \frac{1}{4d(u)} - \sum_{u,w \in M(v)} \frac{1}{2d(u)} \cdot \frac{1}{2d(w)} \\
&= \sum_{u \in M(v)} \frac{1}{2d(u)} \left[\frac{1}{2} - \sum_{w \in M(v)} \frac{1}{2d(w)} \right] \geq \frac{1}{6} \cdot \frac{1}{6} = \frac{1}{36}
\end{aligned}
$$

Luby's Algorithm

By the previous lemma, a good edge will be deleted with probability at least 1/36.

Lemma

At least half of all edges are good.

Proof: Direct every edge towards its endpoint of higher degree, breaking ties arbitrarily.

Claim: For every bad node $v \in V$, there are at least twice as many outgoing edges than incoming edges.

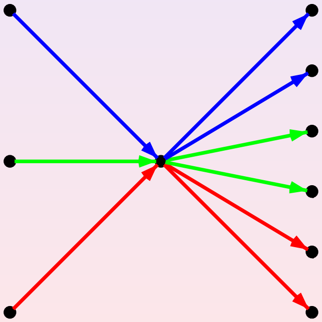
Proof of the claim: Let N_1 be the set of predecessors of v after directing the edges. If $\frac{|N_1|}{d(v)} \geq \frac{1}{3}$, then

$$\sum_{u \in N(v)} \frac{1}{2d(u)} \geq \sum_{u \in N_1} \frac{1}{2d(v)} = \frac{1}{2} \cdot \frac{|N_1|}{d(v)} \geq \frac{1}{6},$$

i.e., v would be good — a contradiction.

Thus, $\frac{|N_1|}{d(v)} < \frac{1}{3}$, i.e., $|N_1| < \frac{1}{2}(d(v) - |N_1|)$, which proves the claim.

Hence, to every bad edge e (for which both endpoints are bad) we can assign a set $P(e) = \{e_1, e_2\}$ of two edges $e_1 \neq e_2$ such that $e \neq f \Rightarrow P(e) \cap P(f) = \emptyset$.



This proves the lemma.

Luby's Algorithm

Theorem

Let X be the number of edges that are deleted (in a certain round). Then for the expected value $\mathcal{E}(X)$ of X we have

$$\mathcal{E}(X) \geq \frac{|E|}{72}.$$

Proof: Let $X_e = 1$, if e is deleted, otherwise $X_e = 0$. Then we have:

$$\begin{aligned} \mathcal{E}(X) &= \sum_{e \in E} \mathcal{E}(X_e) \geq \sum_{e \text{ good}} \mathcal{E}(X_e) \\ &\geq \sum_{e \text{ good}} \frac{1}{36} \geq \frac{|E|}{2} \cdot \frac{1}{36} \end{aligned}$$

Luby's Algorithm

Let m be the total number of edges in our graph. For $i \geq 0$ we define

- $S_i =$ number of edges that were removed in round $1 \cdots i$.
- $X_i =$ number of edges that are removed in round i.

Thus, $S_0 = 0, S_i \leq m$, and $S_{i+1} = S_i + X_{i+1}$.

The statement of the previous theorem can be restated as follows, where $\varepsilon = \frac{1}{72}$:

$$\mathcal{E}(X_{i+1} \mid S_i = \ell) = \sum_{k \in \mathbb{N}} k \cdot \Pr(X_{i+1} = k \mid S_i = \ell) \geq \varepsilon(m - \ell)$$

Lemma

$\mathcal{E}(X_{i+1}) \geq \varepsilon \cdot m - \varepsilon \cdot \mathcal{E}(S_i)$

Luby's Algorithm

Proof:

$$
\begin{aligned}
\mathcal{E}(X_{i+1}) &= \sum_{k \in \mathbb{N}} k \cdot \Pr(X_{i+1} = k) \\
&= \sum_{k,\ell \in \mathbb{N}} k \cdot \Pr(X_{i+1} = k \wedge S_i = \ell) \\
&= \sum_{k,\ell \in \mathbb{N}} k \cdot \Pr(X_{i+1} = k \mid S_i = \ell) \cdot \Pr(S_i = \ell) \\
&= \sum_{\ell \in \mathbb{N}} \Pr(S_i = \ell) \sum_{k \in \mathbb{N}} k \cdot \Pr(X_{i+1} = k \mid S_i = \ell) \\
&= \sum_{\ell \in \mathbb{N}} \Pr(S_i = \ell) \cdot \mathcal{E}(X_{i+1} \mid S_i = \ell) \\
&\geq \sum_{\ell \in \mathbb{N}} \Pr(S_i = \ell) \cdot \varepsilon(m - \ell) \\
&= \varepsilon \cdot m - \varepsilon \cdot \sum_{\ell \in \mathbb{N}} \ell \cdot \Pr(S_i = \ell) = \varepsilon \cdot m - \varepsilon \cdot \mathcal{E}(S_i)
\end{aligned}
$$

Luby's Algorithm

Lemma

$\mathcal{E}(S_i) \geq m(1-(1-\varepsilon)^i)$.

Proof: Induction on i.

The case $i = 0$ is clear.

For $i+1$ we obtain:

$$\begin{aligned}
\mathcal{E}(S_{i+1}) &= \mathcal{E}(S_i) + \mathcal{E}(X_{i+1}) \\
&\geq \mathcal{E}(S_i) + \varepsilon m - \varepsilon \cdot \mathcal{E}(S_i) \\
&= \varepsilon m + (1-\varepsilon)\mathcal{E}(S_i) \\
&\geq \varepsilon m + m(1-\varepsilon)(1-(1-\varepsilon)^i) \\
&= m(1-(1-\varepsilon)^{i+1})
\end{aligned}$$

Luby's Algorithm

Lemma

$\mathcal{E}(S_i) \leq m - 1 + Pr(S_i = m)$

Proof:

$$\begin{aligned}
\mathcal{E}(S_i) &= \sum_{j=0}^{m} j \cdot \Pr(S_i = j) \\
&\leq \sum_{j=0}^{m-1} (m-1) \cdot \Pr(S_i = j) + m \cdot \Pr(S_i = m) \\
&= m \cdot \Pr(S_i = m) + (m-1)(1 - \Pr(S_i = m)) \\
&= m - 1 + \Pr(S_i = m)
\end{aligned}$$

Luby's Algorithm

By the two previous lemmas we have $\Pr(S_i = m) \geq 1 - m(1-\varepsilon)^i$.

Thus, $\Pr(S_i < m) \leq m(1-\varepsilon)^i$.

Choose $k \in O(\log m)$ such that $m(1-\varepsilon)^k \leq 1$.

Then, for $i \geq k$ we have $\Pr(S_i < m) \leq m(1-\varepsilon)^i \leq (1-\varepsilon)^{i-k}$.

Define $f : \mathbb{N} \to \{0,1\}$ by

$$f(x) = \begin{cases} 1 & \text{if } x < m \\ 0 & \text{otherwise.} \end{cases}$$

Thus, $\mathcal{E}(f(S_i)) = \Pr(S_i < m) \leq (1-\varepsilon)^{i-k}$ for $i \geq k$.

Luby's Algorithm

The random variable $R = f(S_0) + f(S_1) + f(S_2) + \cdots$ counts the number of rounds in Luby's algorithm.

We have

$$\begin{aligned}
\mathcal{E}(R) &= \sum_{i \geq 0} \mathcal{E}(f(S_i)) \leq k + \sum_{i \geq k} \mathcal{E}(f(S_i)) \\
&\leq k + \sum_{i \geq k} (1 - \varepsilon)^{i-k} = k + \frac{1}{\varepsilon} \in O(\log m)
\end{aligned}$$

We have shown

Theorem

The expected number of rounds in Luby's algorithm is in $O(\log m)$.

Luby's Algorithm

In the current version of Luby's algorithm we put a node v into S with probability $\frac{1}{2d(v)}$.

For this we have to flip $n = |V|$ many biased coins (with $\Pr(\text{head}) = \frac{1}{2d(v)}$ and $\Pr(\text{tail}) = 1 - \frac{1}{2d(v)}$) independently.

It can be shown that $n^{\Omega(1)}$ many truely random bits (fair coin flips) are necessary (and sufficient) in order to approximate these n independent biased coin flips sufficiently good.

But: In the analysis of Luby's algorithm, we only used *pairwise independence*.

We will show that $\Omega(\log(n))$ many random bits suffice in order to generate $n = |V|$ biased coin flips ($\Pr(\text{head}) \approx \frac{1}{2d(v)}$) that are pairwise independent.

Luby's Algorithm

This leads to a *derandomized version* of Luby's algorithm:

Assume that $\alpha \log(n)$ random bits suffice in order to generate n biased coin flips, where α is a constant. Let $R = \{0,1\}^{\alpha \log(n)}$, thus $|R| = n^\alpha$.

A single round in Luby's algorithm is replaced by the following procedure:

for all $s = a_1 \cdots a_{\alpha \log(n)} \in R$ **do in parallel**
  simulate the next round of Luby's algorithm deterministically
  with $a_i =$ the i-th random bit
endfor
choose that simulation that removes the largest number of edges and go with the resulting graph to the next round

Luby's Algorithm

For every $v \in V$ let $\delta(v) \in \mathbb{R}$ such that

$$\frac{7}{9} \cdot \frac{1}{2d(v)} = \frac{1}{2d(v)} - \frac{1}{9d(v)} \leq \frac{1}{2\delta(v)} \leq \frac{1}{2d(v)}$$

First, we check that the analysis of Luby's algorithm also works when we replace $d(v)$ by $\delta(v)$ everywhere (in particular, $\Pr(v \in S) := \frac{1}{2\delta(v)}$).

Lemma 1 ($\forall v \in V : \Pr(v \in I) \geq \frac{1}{4\delta(v)}$): ✓

Lemma 2 (v good $\Rightarrow \Pr(v \in N(I)) \geq 1/36$): ✓

Recall that v is good if $\sum_{u \in N(v)} \frac{1}{2\delta(u)} \geq \frac{1}{6}$ and that an edge is good if one of its endpoints is good.

Luby's Algorithm

Instead of showing that at least half of the edges are good (Lemma 3), we will prove that at least 1/4 of all edges are good.

Again, we direct every edge towards its endpoint with larger δ-value.

Lemma

Let $n_1 = |N_1|$ be the number of incoming edges of a node v. If $\frac{7n_1}{18d(v)} \geq \frac{1}{6}$ then v is good.

Proof

$$\begin{aligned} \sum_{u \in N(v)} \frac{1}{2\delta(u)} &\geq \sum_{u \in N_1} \frac{1}{2\delta(v)} = \frac{n_1}{2\delta(v)} \\ &\geq \frac{7}{9} \cdot \frac{n_1}{2d(v)} \geq \frac{1}{6} \end{aligned}$$

Luby's Algorithm

Thus, if v is bad then $\frac{7}{18} \cdot \frac{n_1}{d(v)} \leq \frac{1}{6}$.

We obtain $n_1 \leq \frac{3}{7} \cdot d(v)$.

Thus, $d(v) - n_1 \geq \frac{7}{3} n_1 - n_1 = \frac{4}{3} n_1$.

Therefore, $n_1 \leq \frac{3}{4}(d(v) - n_1)$, i.e., at least $1/4$ of all edges is good.

If X is the number of edges that are removed (in a certain round), then we obtain

$$\mathcal{E}(X) \geq \frac{1}{36} \cdot \frac{|E|}{4} = \frac{1}{144} |E|$$

Luby's Algorithm

We have shown that Luby's algorithm works with $\delta(v)$ instead of $d(v)$ as well.

Recall δ only has to satisfy $\dfrac{1}{2\delta(v)} \in \left[\dfrac{7}{9} \cdot \dfrac{1}{2d(v)}, \dfrac{1}{2d(v)}\right]$.

Now choose a prime number p with $9n \leq p \leq 18n$ — such a prime exists by Betrand's postulat. We may identify V with a subset of $\mathbb{F}_p = \{0, \dots, p-1\}$.

The interval $\left[\frac{7}{9} \cdot \frac{1}{2d(v)}, \frac{1}{2d(v)}\right]$ has size $\frac{1}{2d(v)} - \frac{7}{9} \cdot \frac{1}{2d(v)} = \frac{1}{9d(v)} \geq \frac{1}{9n} \geq \frac{1}{p}$, thus there exists a number of the form $\frac{a_v}{9n}$ in this interval for some $a_v \in \mathbb{N}$. We can set $\frac{1}{2\delta(v)} = \frac{a_v}{p}$.

Determine a subset $A_v \subseteq \mathbb{F}_p$ with $|A_v| = a_v$, where $\frac{7}{9} \cdot \frac{1}{2d(v)} \leq \frac{a_v}{p} = \frac{1}{2\delta(v)} \leq \frac{1}{2d(v)}$

Luby's Algorithm

Now choose $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ randomly (using $\mathcal{O}(\log(n))$ many random bits) and put v into S if and only if $x + vy \in A_v$.

Since for every $y, z \in \mathbb{F}_p$ there is exactly one $x \in \mathbb{F}_p$ with $x + vy = z$, namely $x = z - vy$, we have

$$\begin{aligned}
\Pr(v \in S) &= \frac{1}{p^2} |\{(x, y) \mid x + vy \in A_v\}| \\
&= \frac{1}{p^2} \sum_{z \in A_v} |\{(x, y) \mid x + vy = z\}| \\
&= \frac{1}{p^2} \sum_{z \in A_v} p \\
&= \frac{a_v}{p}
\end{aligned}$$

Luby's Algorithm

Finally, we have to show pairwise independence: Let $u \neq v$ be two different nodes. Then

$$\Pr(u \in S \wedge v \in S) = \frac{1}{p^2} |\{(x,y) \mid x + uy \in A_u \wedge x + vy \in A_v\}|$$

$$= \frac{1}{p^2} \sum_{(z_u, z_v) \in A_u \times A_v} \left| \left\{ (x,y) \mid \begin{pmatrix} 1 & u \\ 1 & v \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} z_u \\ z_v \end{pmatrix} \right\} \right|$$

Luby's Algorithm

The matrix has an inverse (the determinant is $v - u \neq 0$), thus for every (z_u, z_v) there is exactly one $(x, y) \in \mathbb{F}_p \times \mathbb{F}_p$ for

$$\begin{pmatrix} 1 & u \\ 1 & v \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} z_u \\ z_v \end{pmatrix}$$

We obtain

$$\Pr(u \in S \wedge v \in S) = \frac{1}{p^2} a_u a_v = \frac{a_u}{p} \frac{a_v}{p} = \Pr(u \in S) \cdot \Pr(v \in S).$$

We have shown pairwise independence.