Parallel Expression Evaluation

Example Input:

```
+    
/   
+   +
/ 
X  X
/ 
1 4 5 3
```

Output! Value, all subvalues

Goal! Parallel Alg

Simple Alg

1) Assign a processor to each node.
2) While tree non empty do
   2a) if leaf send value to parent
        delete node [RAKE]
   2b) if node has 2 values then evaluate
Worst Case for Simple Alg

Recall: Horner's Rule

Input: polynomial $a_0 + a_1x + a_2x^2 + \ldots + a_nx^n$

Alg: $a_0 + x(a_1 + x(\ldots + x(a_{n-1} + x(a_n))\ldots))$

As a tree

```
    +
   / |
  +  |
  /   |
  x   +
   /   |
   x   |
    /   |
   a_n |
```

Simple Alg

$O(n)$ Time

$O(n^2)$ Work
Keeping nodes with only one value busy!

Here we view the tree edges as transformers.

Init: The edge are the identity.

\[ a(y+b) = ay + ab \]
The general case.

\[ f(y) = ay + b \quad g(y) = cy + d \]

\[ f(g(y)) = a(cy + d) + b = acy + (ad + b) \]

Note: fons \( ay + b \) are closed under compositions

We can also remove an independent set of 1-child nodes (degree 2 nodes)

Very similar to pivoting in Gaussian Elim!
**Def** \( V_0 \ldots V_K \) is a chain if:

1) \( V_{i+1} \) is only child of \( V_i \) \( 0 \leq i < K \).
2) \( V_K \) has only one child & it is not a leaf

**Eg:**

![Diagram showing a chain](image)

---

**The Independent Set**

1) All leaves
2) Max independent set from each maximal chain
Parallel Tree Contraction

RAKE $\equiv$ remove all leaves

COMPRESS $\equiv$ replace each maximal chain of length $K$ with one of length $\sqrt[3]{2}$.

\[ \text{CONTRACT} = \{ \text{RAKE}, \text{COMPRESS} \} \]

Thm $\mid \text{CONTRACT}(T) \mid \leq \frac{2}{3} \mid T \mid$

pf

Def $V_0 = \text{leaves of } T$

$V_1 \subseteq V$ with 1 child

$V_2 \subseteq V$ with $2 \leq \# \text{ children}$

$C \subseteq V_1$ with child in $V_0$
Claim: $|V_0| > |V_a|$

Proof: Induct on size of $T$.

Claim: $|V_0| > |C|$.

**Def** $R_a = V_0 \cup V_2 \cup C$

$C_{om} = V_1 - R_a$

$|\text{RAKE}(R_a)| \leq |V_2 \cup C|$ $\Rightarrow |\text{RAKE}(R_a)| \leq \frac{2}{3}|R_a|$

*Note:* $C_{om} = \text{union of maximal chains}$

$|\text{COMPRESS}(C_{om})| \leq \frac{1}{2}|C_{om}|$

Cor: After $\log_{3\cdot 2} n$ CONTRACTS Two empty
Work and Time Efficient Tree Contraction

Idea 1: Do regular RAKE.
Use Random-Mate to COMPRESS chains.

Using Chernoff Bounds

Thm: Randomized Tree Contraction runs in $O(\log n)$ with high prob.

Thus $W = O(n \log n)$, $\text{Time} = O(\log n)$.

Idea 2: 1) Break tree in $n/\log n$ pieces each of size $\log n$.
2) Contract pieces to constant size.
3) Run Rand Tree Contraction on tree of size $O(n/\log n)$. 
A Tree into Bridges

Let $T$ be a rooted tree $T = (V, E)$

Def A subtree $B$ is a bridge if at most 2 attachments: a root, leaf.

Ex 1) Single edge
2) Induced subtree
3) 

Thm $\forall m \in \text{decomp of } T \text{ into } O(n/m)$ bridges of size $m$. 
\[ T = (V, E) \quad W(v) = \text{number of nodes in subtree rooted at } V \]

**Def** \( V \) is \( m \)-critical if

1. \( V \) not a leaf.
2. \( \frac{rw(v)}{m} > \frac{rw(v')}{m} \) \( V' \) child of \( V \).

**e.g.** 5-critical

**5-bridges**

Claim (see chap 3) \( (m-1) \)-bridges proves thm.
Thin Tree Contraction can be done in \( O(n) \) work \( O(\log n) \) time with high prob.

Known Det in same bounds.

Alg

1) Compute \( \log n \)-critical nodes using Euler tour
2) Contract bridges
3) Contract \( \frac{1}{\log n} \) tree using random mate
4) Expand.