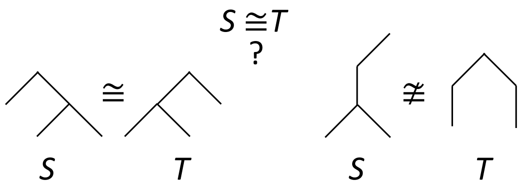
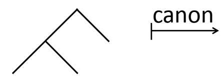
Binary Tree Isomorphism

Given two binary trees, are they isomorphic as directed graphs?



Canonization

Assign to each binary tree *T* a unique isomorphism invariant name for *T*:



[• [• •]]

string

Previous Work

- [Aho, Hopcroft, Ullman, 1974]
  Bottom-up vertex refinement — $O(n)$ time
- [Ruzzo, 1981]
  Auxilary logspace PDA — NC for $O(\log n)$ degree
- [Miller, Reif, 1991]
  tree contraction — $O(\log n)$ time

Definitions

Tree $T = \langle V, E\rangle$ labels $V = \{1, \ldots, n\}$ edges $E \subseteq V^2$

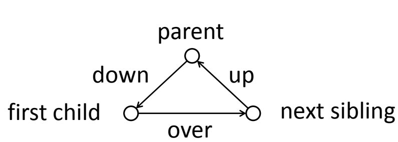
$\langle V, E\rangle = S \cong T = \langle V, E'\rangle$ (n.b. same domain)

If and only if there is a permutation $\alpha : V \longrightarrow V$, $\alpha(v) = \hat{v}$ ∍

$$u \xrightarrow{E} v \iff \hat{u} \xrightarrow{E'} \hat{v}$$

In addition to the edge relation $u \longrightarrow v$, we also use the label ordering $u < v$ (but final answer is invariant of <).

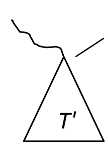
Logspace Tree Traversal



state	last move	next attempt
new	down, over	down, over, up
old	up	over, up

Subtree Cardinality

To compute cardinality $|T'|$:



subtree

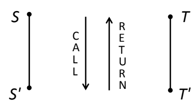
initial position
state = new
count := 1

→ traverse (one move)
if state = new then
  count := count +1
if position = initial then
  terminate
else

Isomorphism Test

A. Check sizes: $|S| = |T| = n$
B. Check children: $\# S = \# T = k$
C. $k = 0$ Ans $\longleftarrow$ YES $S = \cdot \cong \cdot = T$
$k = 1$ Recursion w/o Stack

CALL
move pebbles
down and
GOTO entry

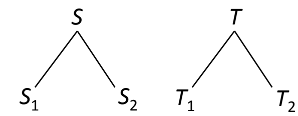


RETURN
move pebbles
up and
GOTO exit

- 2-pronged traversal: local variables are pebbles
- program counter is computed: deduce return address by examining current and previous positions

Key Idea

$k = 2$: Check that all sizes match



- If $|S_1| = |T_1| < |S_2| = |T_2|$

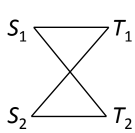
$S_1 \cong T_1$? $\xrightarrow{\text{yes}}$ $S_2 \cong T_2$?

remember last position upon return, to distinguish between larger/smaller pair

Equicardinality Case

- $|S_1| = |T_1| = |S_2| = |T_2| < {}^{n}/_{2}$

Use $O(1)$ bits to sequence calls and store answers of 4 cross-comparisons



(1, 1)	(1, 2)	(2, 1)	(2, 2)	
$\cong$	$\cong$	$\cong$	$\cong$	√
$\cong$	$\ncong$	$\ncong$	$\cong$	√
$\ncong$	$\cong$	$\cong$	$\ncong$	√

Results

Theorem: Directed Tree Isomorphism is in DLOGSPACE
Theorem: Directed Tree Canonization is in DLOGSPACE (write-only output tape)

"Improves" best known results (but inefficient in time complexity, due to extensive recomputation)

L = $O(\log n)$ space $\subseteq$ $O(\log n)$ time on EREW PRAM

L $\subsetneq$ P ?!

Logical Significance

Original Goal: Shows that the *logspace computable intrinsic properties* of trees are recursively indexable.
Provides (indirectly) a *logic* for Lspace = Q(L) on trees.

- Graphs $\underbrace{\langle V, E\rangle}_{} \overset{?}{\cong} \underbrace{\langle V, E'\rangle}_{}$ $\in$ P?
- Trees $S \overset{?}{\cong} T$ $\in$ L
- Strings $\underbrace{\langle V, <, (bit), u\rangle}_{w} \overset{?}{\cong} \underbrace{\langle V, <', (bit'), u'\rangle}_{w'}$ $\in O(1)$ time on WRAM, with $w \overset{?}{=} w'$

Extensions & Future Directions

- Results extend to *undirected* trees, and *forests*
- Tree Canonization $\in NC^1$?
  - **Conjecture**: [Buss, Lindell] Binary Tree isomorphism $\in NC^1$, given transitively-closed trees.
- Is there a logic for Ptime = Q(P) on graphs?
  - **Conjecture**: [Gurevich] No