Binary Tree Isomorphism

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

\[ S \cong T ? \]

\[ S \not\cong T \]

Canonization

Assign to each binary tree \( T \) a unique isomorphism invariant name for \( T \):

\[ \text{tree} \mapsto \left[ \bullet \left[ \bullet \bullet \right] \right] \]

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 \( T = (V, E) \) \( V = \{1, \ldots, n\} \) edges \( E \subseteq V^2 \)

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

If and only if there is a permutation \( \alpha : V \rightarrow V' \), \( \alpha(v) = \hat{v} \appa \)

\( u \longrightarrow v \iff \hat{u} \longrightarrow \hat{v} \)

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

Logspace Tree Traversal

Subtree Cardinality

To compute cardinality \( |T'| \):

\[
\begin{array}{c|c|c}
\text{state} & \text{last move} & \text{next attempt} \\
\hline
\text{new} & \text{down, over} & \text{down, over, up} \\
\text{old} & \text{up} & \text{over, up}
\end{array}
\]

\[
\begin{array}{c}
\text{initial position} \\
\text{state = new} \quad \text{count := 1} \\
\text{if position = initial then} \quad \text{terminate}
\end{array}
\]
Isomorphism Test

A. Check sizes: \(|S| = |T| = n\)
B. Check children: \(#S = #T = k\)
C. \(k = 0\) Ans \(\leftarrow\) YES \(S = \ast \equiv \ast = T\)
   \(k = 1\) Recursion \(w/o\) Stack

Key Idea

\(k = 2: \text{ Check that all sizes match}\)

\[ \begin{array}{c}
S \\
S' \\
\end{array} \quad \begin{array}{c}
T \\
T' \\
\end{array} \]

• If \(|S_1| = |T_1| < |S_2| = |T_2|\)
  \(S_1 \cong T_1\) ? \(\text{yes}\) \(S_2 \cong T_2\) ?

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

Logical Significance

Original Goal: Shows that the logspace computable intrinsic properties of trees are recursively indexable.
Provides (indirectly) a logic for \(LSPACE = \mathcal{Q}(L)\) on trees.
• Graphs \((V, E) \cong (V', E')\) \(\in P?\)
• Trees \(S \cong T\) \(\in L\)
• Strings \((V, <, (bit, u)) \cong (V', <', (bit', u'))\) \(\in O(1)\) time on \(WRAM\)

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

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 \(EREW\) \(PRAM\)

\(L \not\subseteq P \text{?!}\)

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 \(PTIME = \mathcal{Q}(P)\) on graphs?
  → Conjecture: [Gurevich] No