MSO Query Answering on Trees

Marcin Chrzanowski

December 20, 2022

Question Answering

- Fix a function $f : \mathcal{S} \times \mathcal{Q} \to \mathcal{A}$
- *preprocessing*: on input $S \in \mathcal{S}$ output an indexing structure S'.
- *answering*: on input $Q \in \mathcal{Q}$, S' outputs $f(S, Q)$.

Question Answering - Time Complexity

- Let $n = |S|$, $m = |Q|$.
- Preprocessing algorithm works in time $f(n)$
- Answering algorithm works in time $g(n, m)$
- Notate the full algorithm's time complexity as $\langle f(n), g(n, m) \rangle$.

Example: LCA

Least Common Ancestor problem (LCA)

Given: a tree T.
Questions: for two vertices x and y, what is the lowest (furthest from the root) vertex that is an ancestor of both x and y?

Classic Harel and Tarjan result: this can be solved in $\langle O(n), O(1) \rangle$.

MSO Query Answering on Trees

Fixed: an MSO formula $\varphi(\vec{X})$ over trees with k free second-order variables.

Given: a tree T.

Questions: is $\vec{W}$, a k-tuple of subsets of T's vertices, a satisfying assignment to $\vec{X}$? I.e., does $T \models \varphi(\vec{W})$.

Note: first-order variables are supported by restricting input sets to singletons.

Prior Work

Reduce to Model Checking

- Courcelle: MSO model checking over structures of bounded treewidth is $O_\varphi(n)$.
  - MSO query answering in $\langle O(1), O_\varphi(n)\rangle$.
- Amarilli et al.: $O_\varphi(n)$ preprocessing, $O_\varphi(\log n)$ relabeling.
  - MSO query answering in $\langle O_\varphi(n), O_\varphi(m \log n)\rangle$.

Kazana

- In his PhD thesis, solved MSO query answering (there called query *testing*) in $\langle O_\varphi(n), O_\varphi(1)\rangle$.
- But limited to formulae whose free variables are all first-order.
- Uses Colcombet's factorization forests.

Our solution

- I show an $\langle O_\varphi(n), O_\varphi(m \log m)\rangle$ solution to MSO query answering.
- $O_\varphi(1)$ for first-order free variables, matching Kazana's result.
- Should be understandable by a CS student who has taken undergraduate algorithmics and automata theory courses.

Reductions

- Reduce from MSO to tree automaton (nonelementary wrt to φ).
- Transform to a binary tree.

Relabel Regular Questions on Trees

Fixed: a deterministic bottom-up tree automaton A over Σ.

Given: a tree T labeled with Σ.

Questions: given m relabelings $v_1 \mapsto a_1, \ldots, v_m \mapsto a_m$, for $v_i \in V(T)$ and $a_i \in \Sigma$, what state does A arrive at in the root of T', where T' is T with each v_i's label modified to the corresponding a_i?

Answering Questions

Let W be the set of relabeled vertices, $m := |W|$.

We'll partition the tree such that:

- There will be $O(m)$ rooted parts.
- Each element of W will be the root of some part (there may be other parts not rooted in an element of W).
- Working bottom-up, we can compute A's state in the root of each part in $O(1)$.

LCA Closure Questions

Given: a tree T.

Questions: given $W \subseteq V(T)$, output the LCA closure of W.

Can be solved in $\langle O(n), O(m \log m) \rangle$ (Section 4.1.2).

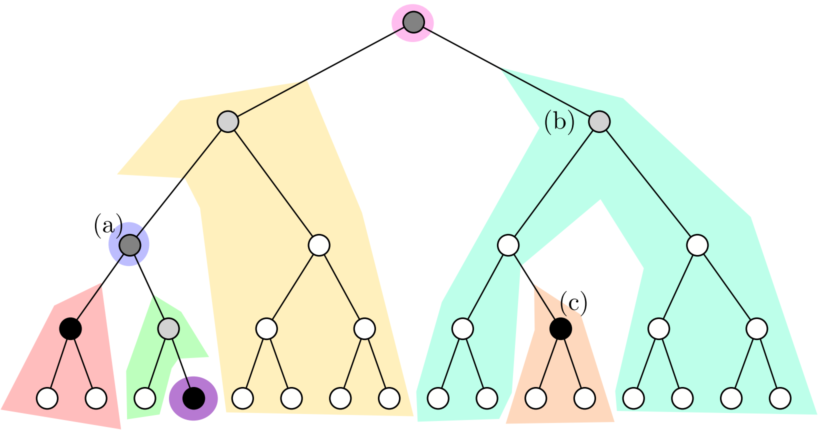
LCA Closure Questions

- Preprocess for LCA.
- Given W, sort it according to in-order numbers.
- Add the LCA's of pairs of subsequent vertices in the sorted list.

Types of Parts

- Subtree.
- Singleton.
- Subtree with a hole.

Types of Parts



Computing Roots of Parts - Subtree

- During preprocessing, we precompute A's run over T.
- In a subtree part, only the root was relabeled.
- Apply A's transition function to the precomputed states of the root's children and the root's new label.

Computing Roots of Parts - Singleton

- Both of the singleton's children are roots of parts.
- We're working bottom-up, we've already computed the states in those roots.
- Apply A's transition function to those and the singleton's new label.

Computing Roots of Parts - Subtree With a Hole

- Nontrivial case.
- Idea: can be computed by a DFA walking up from hole to root.
- Chapter 3: Branch Infix Regular Questions
  - Show how to solve this in $\langle O_A(n), O_A(1) \rangle$.
  - Generalization of a known algorithm on words.

Computing Roots of Parts - Subtree With a Hole

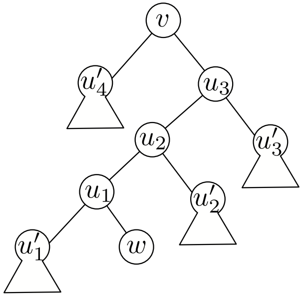


Figure 1: Information needed for subtree of v with hole w

Branch Infix Regular Questions

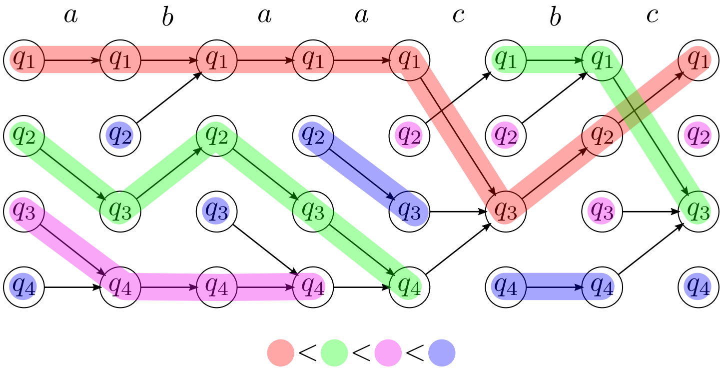
Branch Infix Regular Questions

Fixed: regular language L over alphabet Σ, given by DFA A.

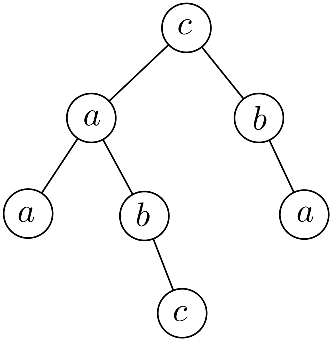
Given: a tree T labeled with Σ.

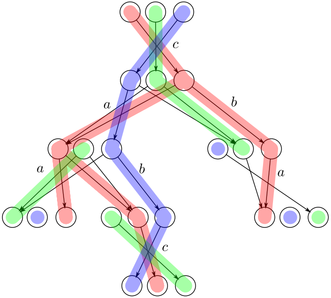
Questions: given a vertex x and its descendant y, does the word given by labels on the path from x to y belong to L?

The Word Case



Generalizing to Trees





Jumping Down in a Tree

- Need to be able to decide which node to jump down to when color path breaks.
- For each color, mark nodes where the color breaks.
- We can compute the highest marked descendant on a path between two nodes in $\langle O(n), O(1) \rangle$
  - Method inspired by RMQ algorithm by Bender and Farach-Colton.

Highest Marked Descendant on Path

- `pre`: pre-order numbers of T's vertices, arranged in post-order.
- `index[`v`]`: v's index in `pre`.
- For x and its descendant y, consider the range `pre[index[`y`], index[`x`] - 1]`:
  - All the values correspond to descendants of x.
  - Values smaller than `pre[index[`y`]]` correspond to ancestors of y.
- For unmarked nodes, set their value in `pre` to ∞.
- Now a range minimum query over the above range gives us the answer.

Highest Marked Descendant on Path

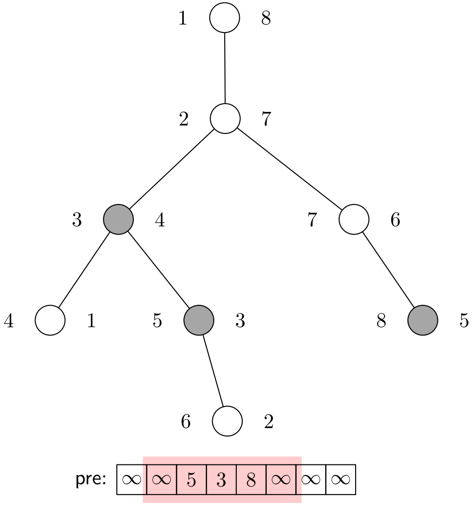


Figure 2: `pre` for example tree