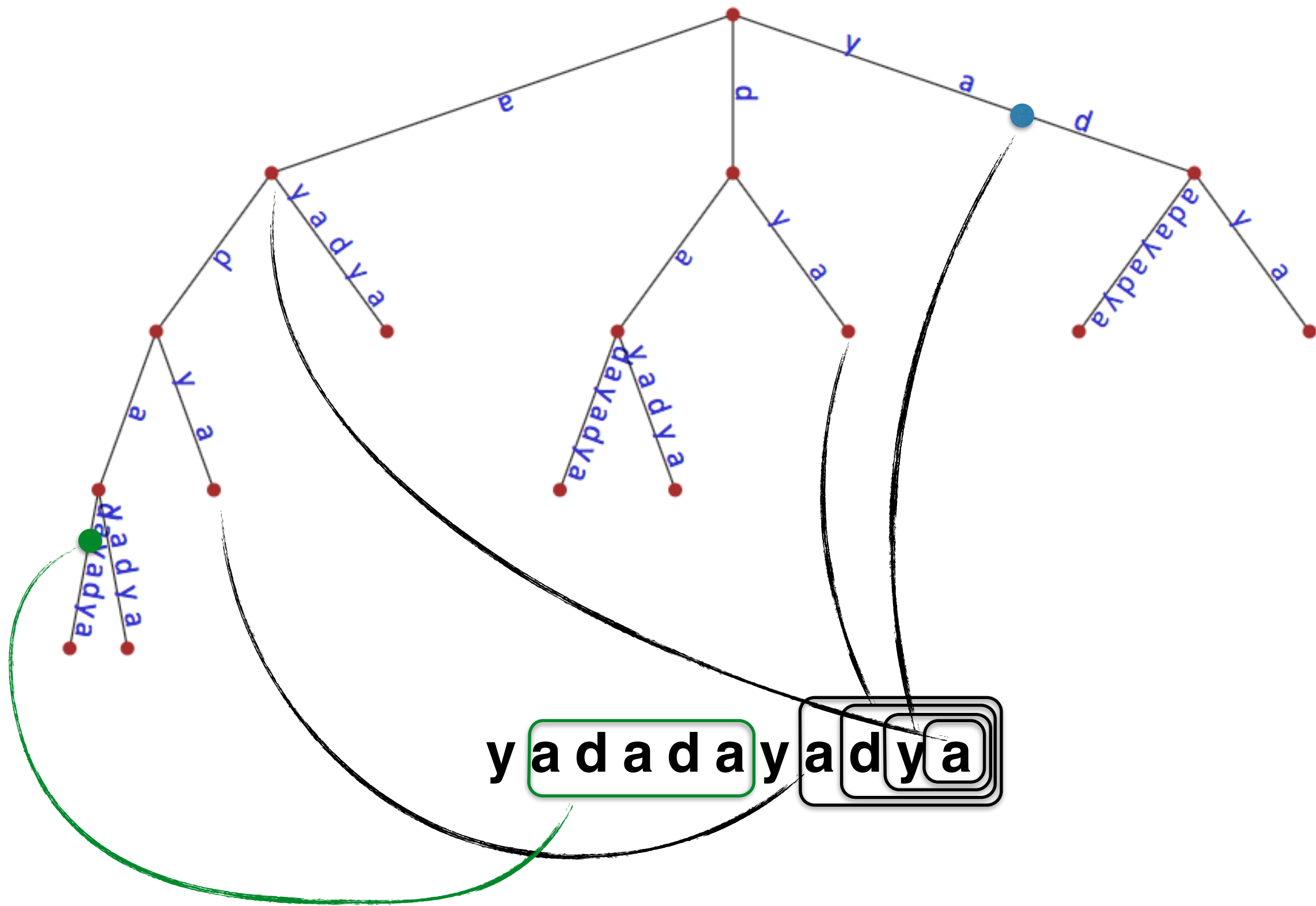
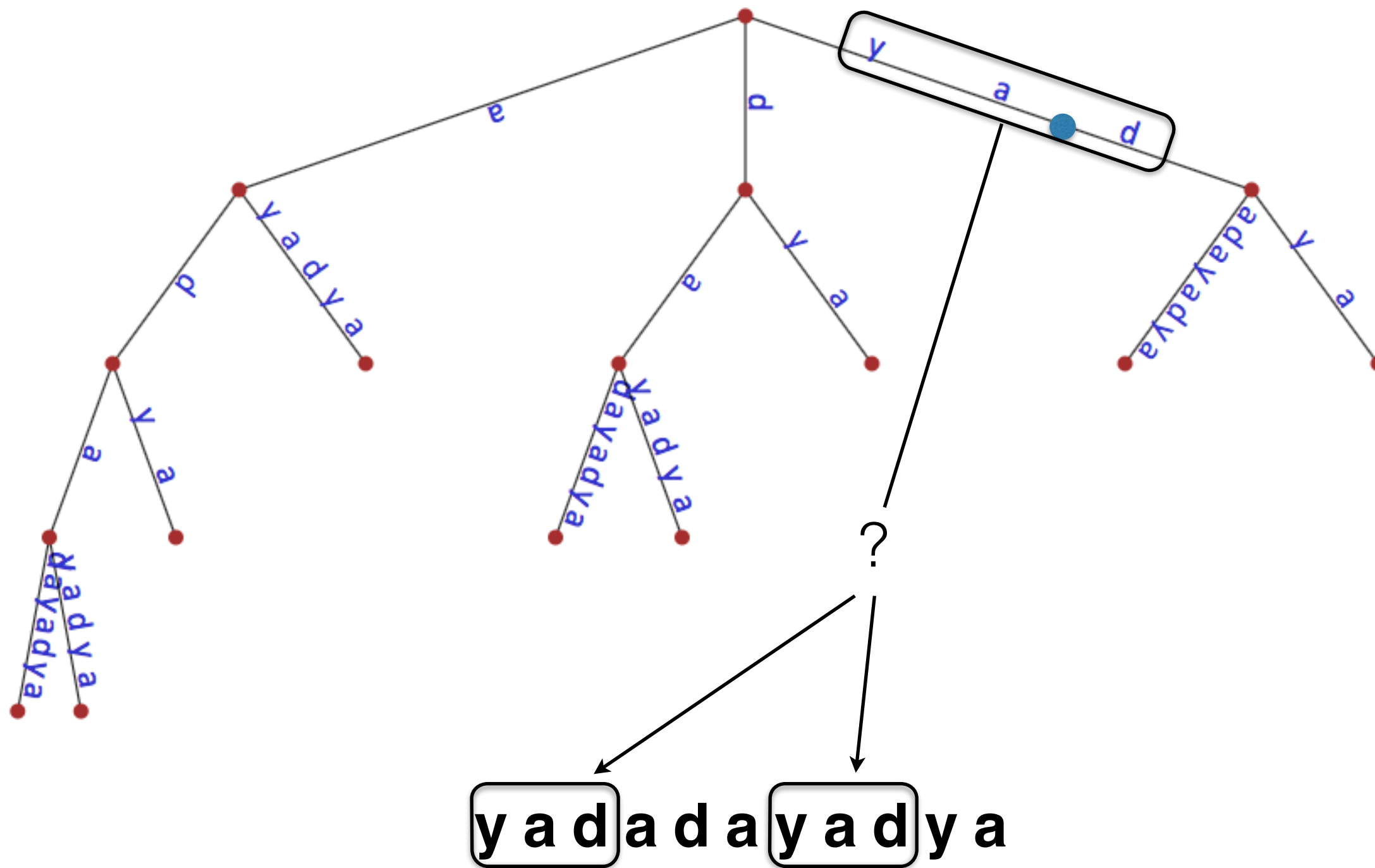
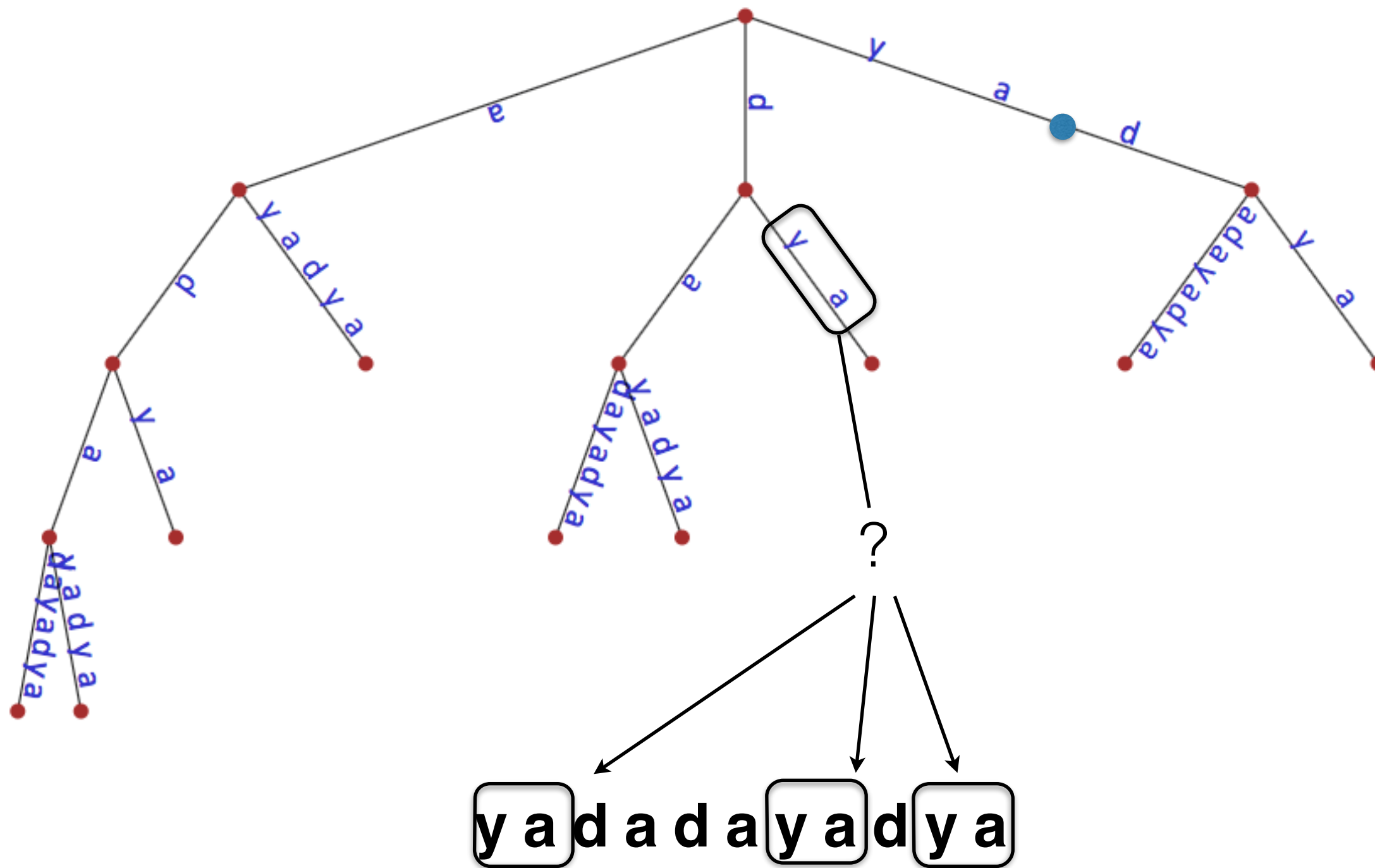


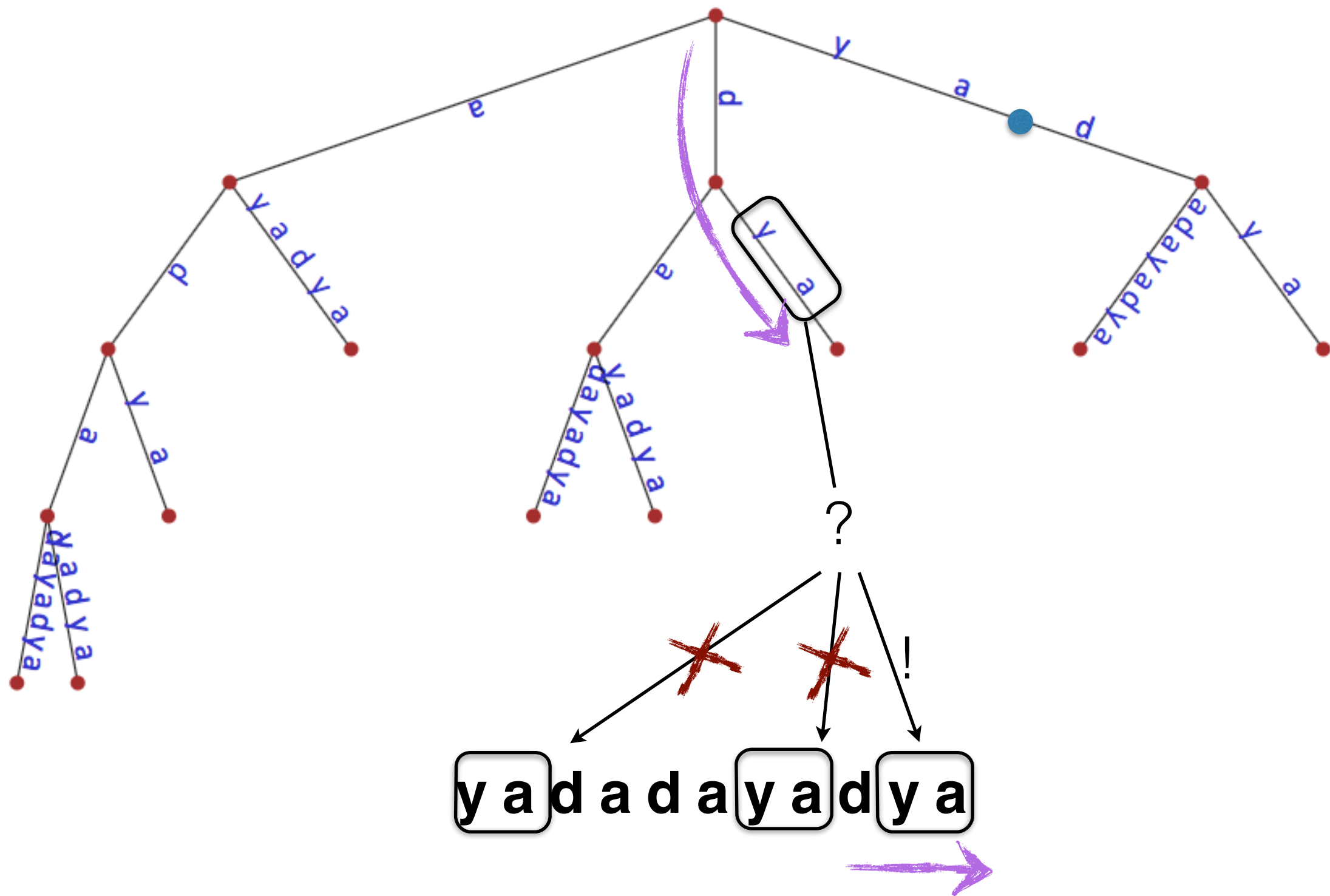
Most Recent Match Queries in On-Line Suffix Trees

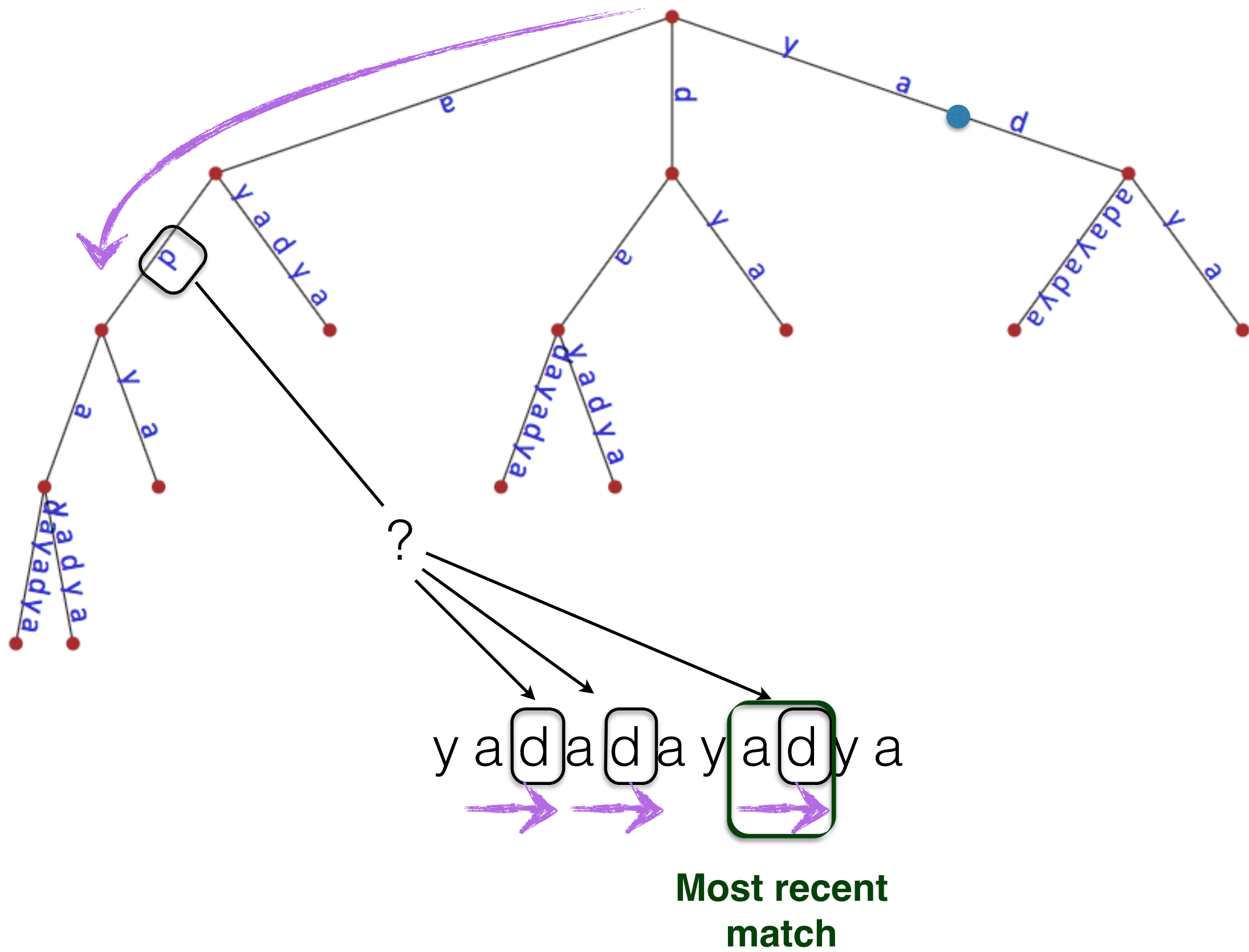
N. Jesper Larsson



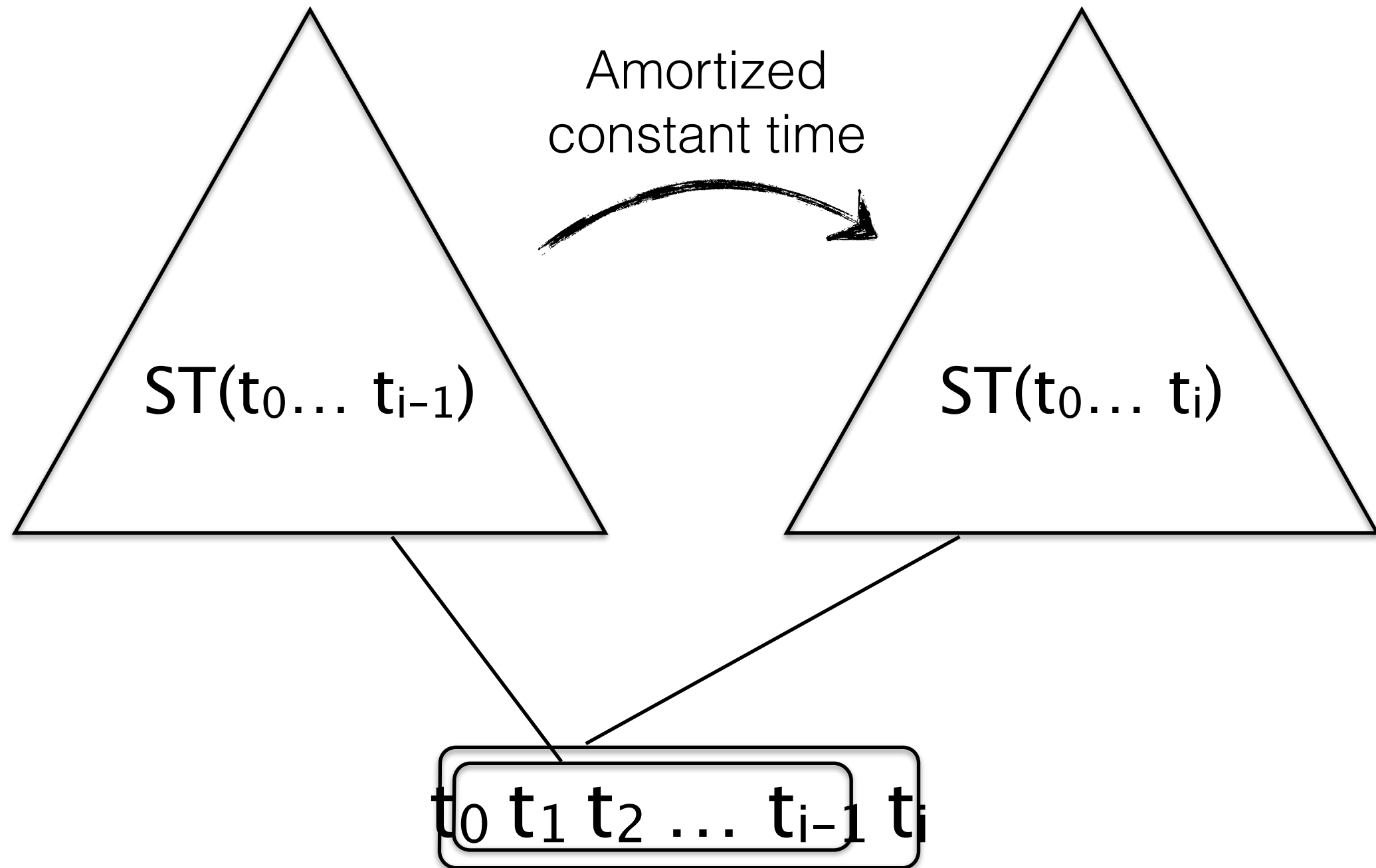






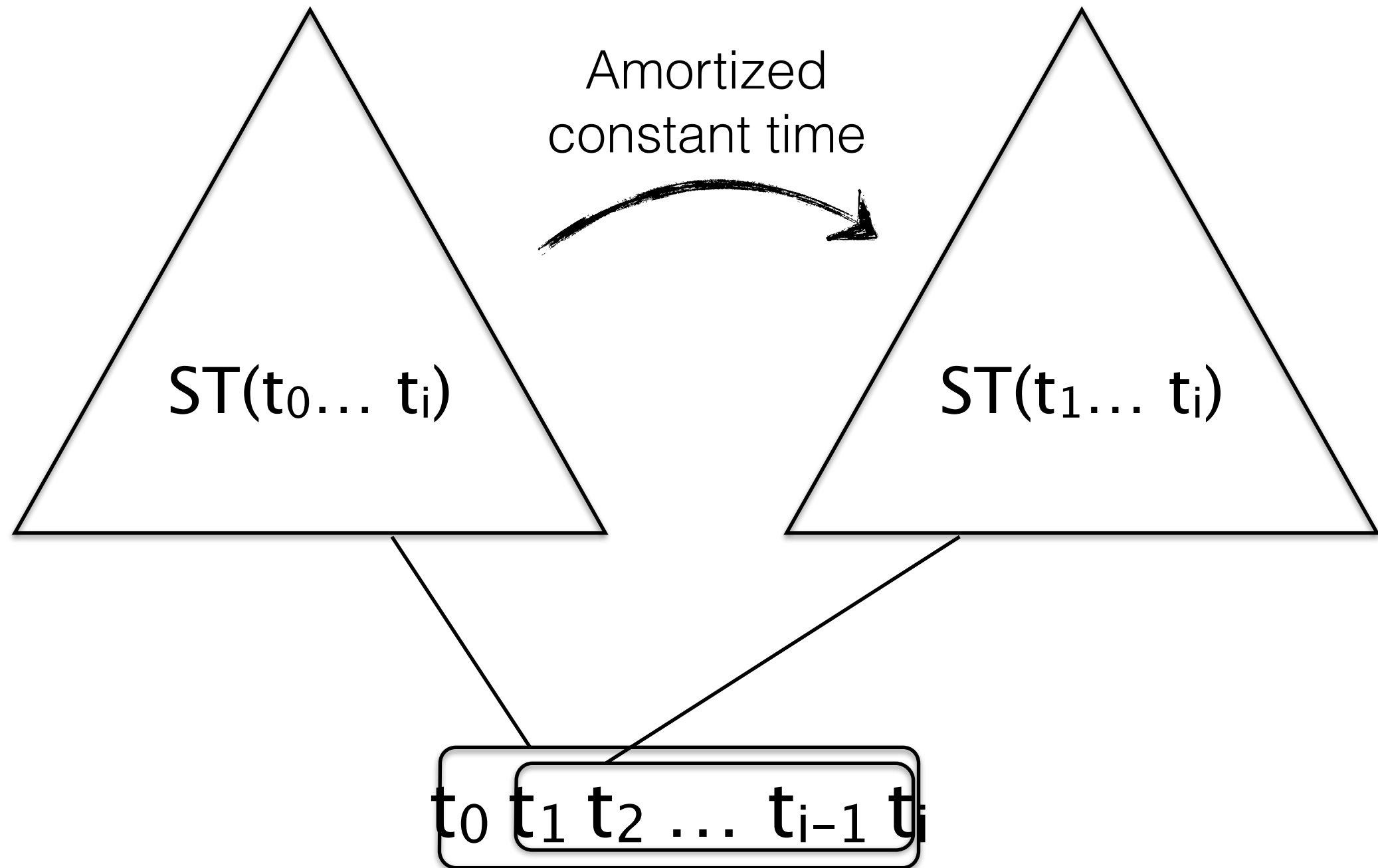


On-Line Ukkonen 1995



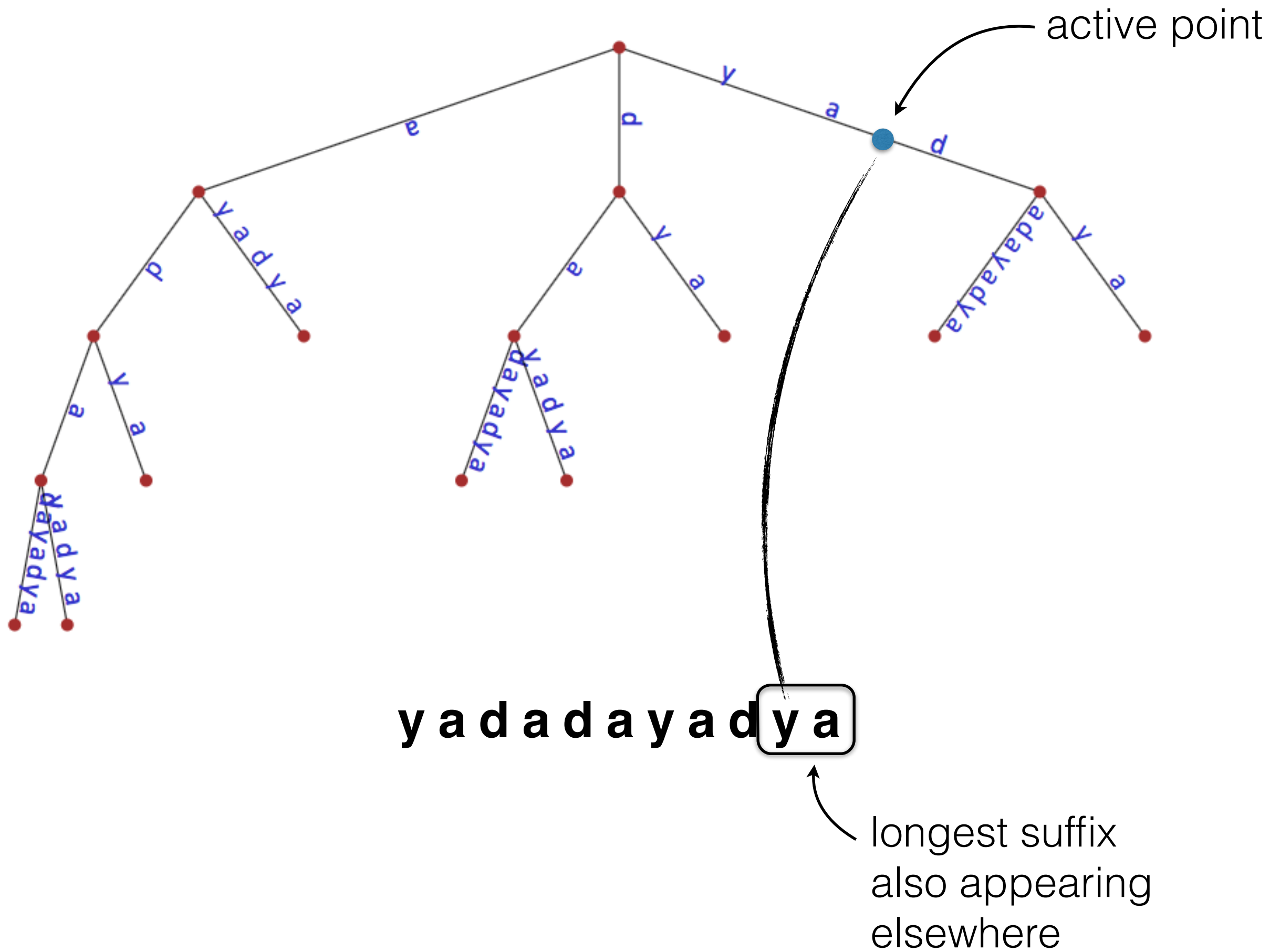
Sliding Window

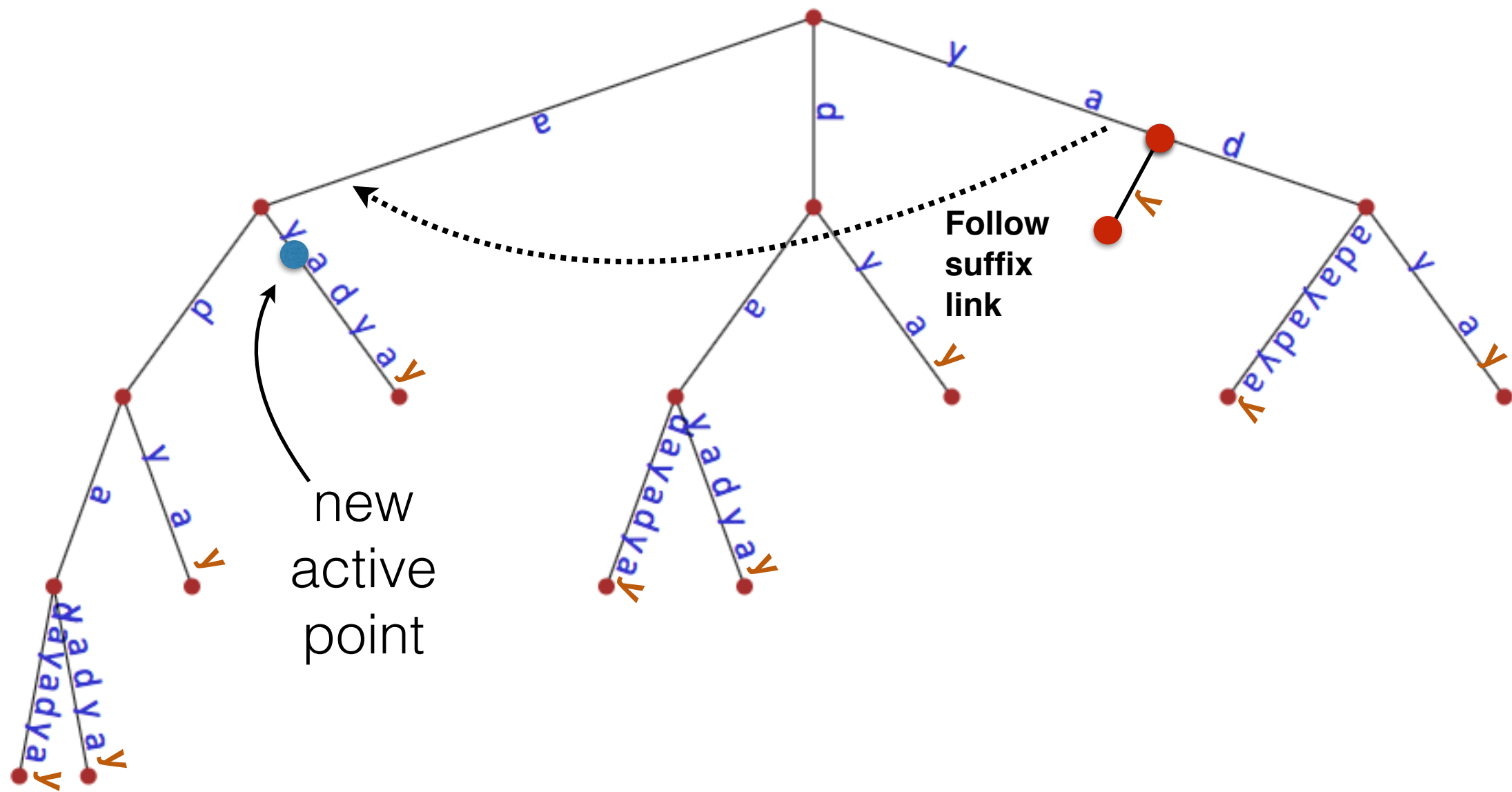
Fiala & Greene 1989 (+ Larsson 1996)



Previous results

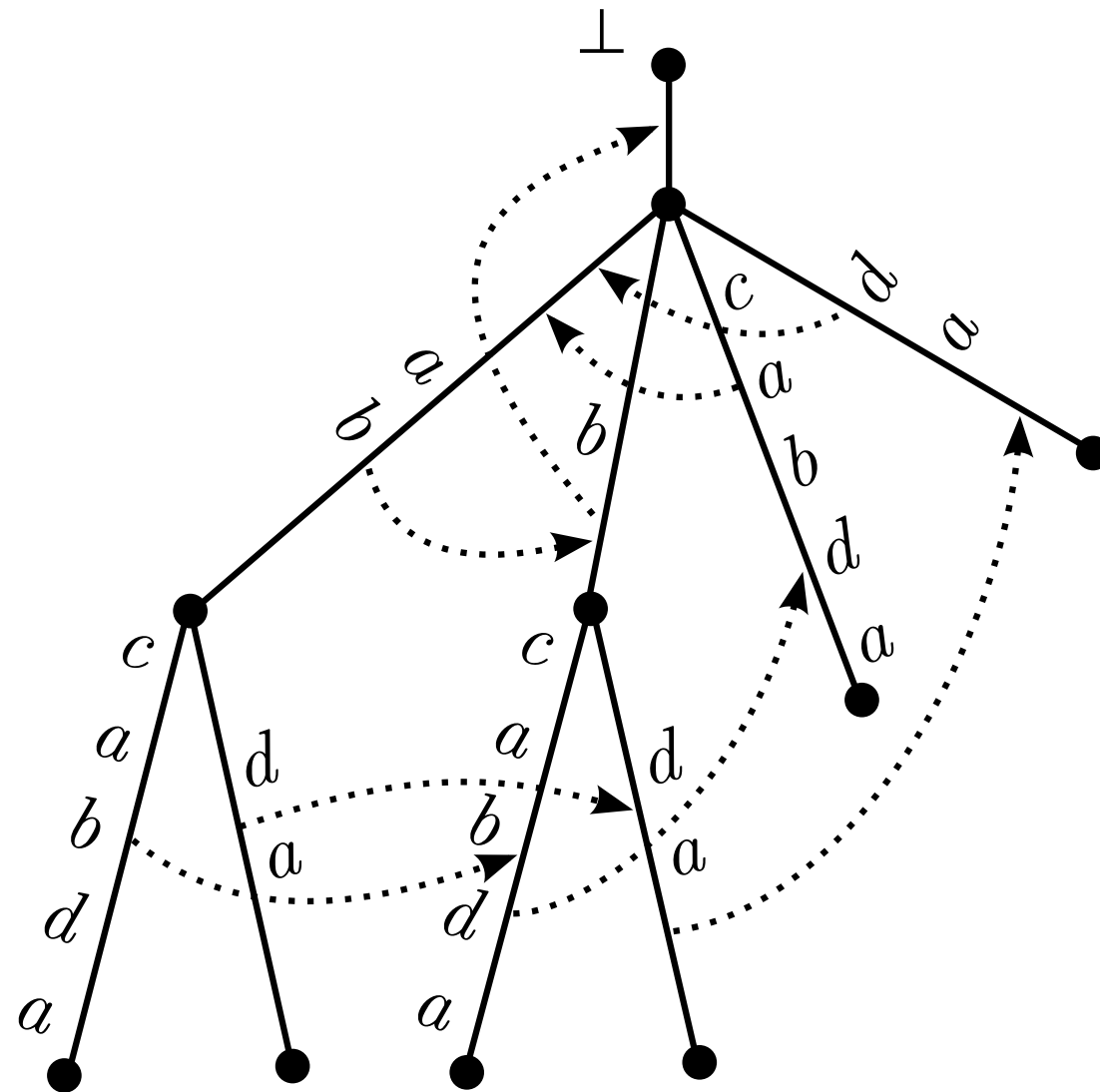
- Amir, Landau, Ukkonen 2002: $O(N \log N)$, when pattern is suffix of currently indexed string
- Ferragina, Nitto, Venturini 2009: $O(N)$ for off-line special case LZ factorization
- Chochmore, Langiu, Mignosi 2013: $O(N)$ for *equal-cost* problem (not guaranteeing most recent match), under certain assumptions of entropy coding





yadadayaya

Suffix links



⊥

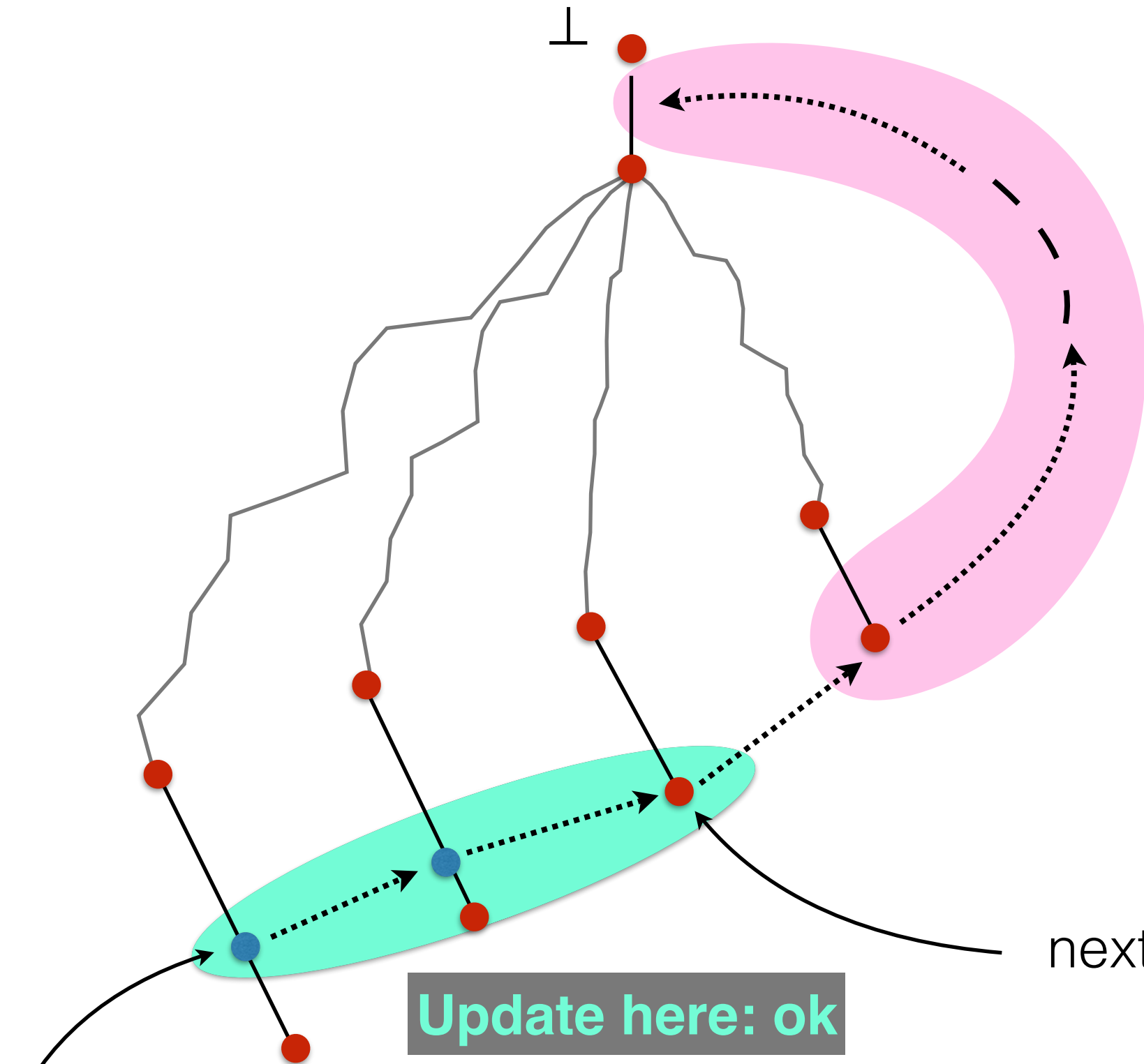
Update all:
superlinear

Traverse after
matching P: IPI
ok

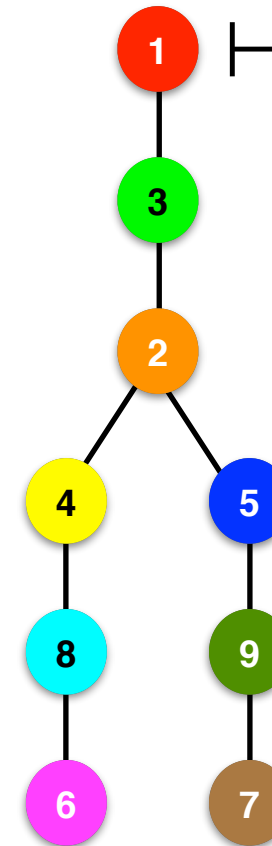
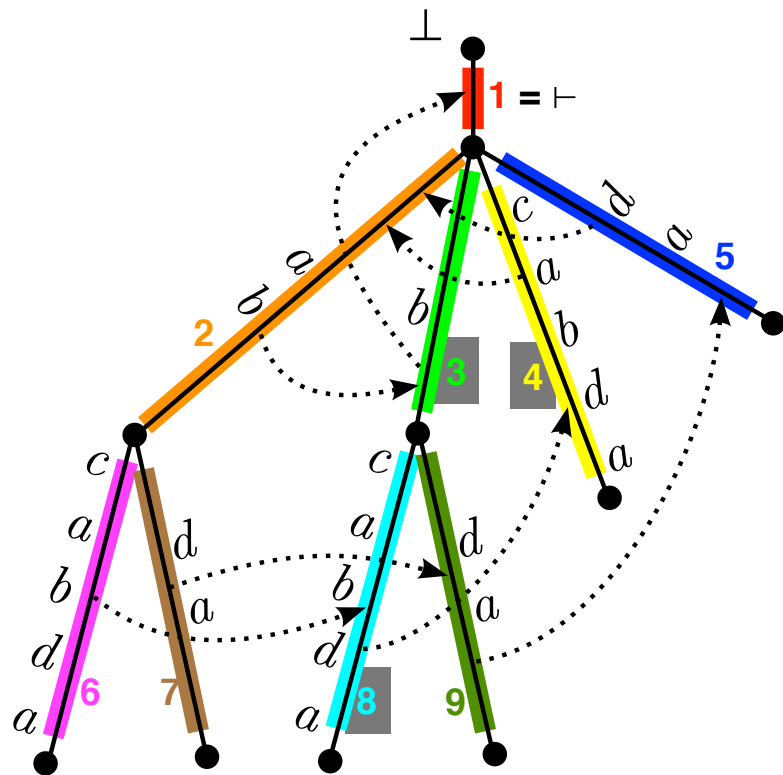
next active point

Update here: ok

active point



Link Tree



Property 1

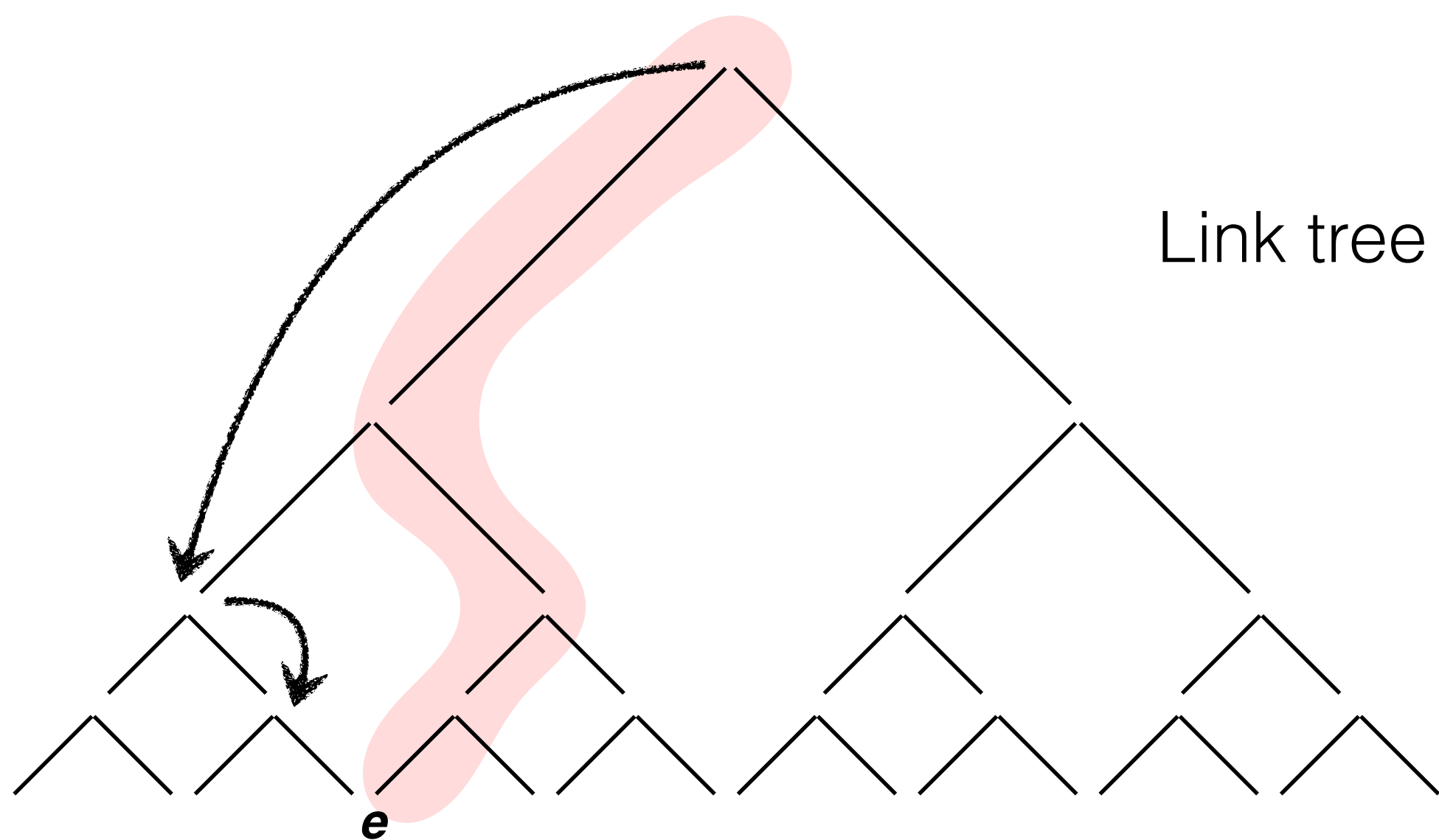
- When active point reaches end of edge, *pos-update* that edge
- For every **link tree** node ***e***, maintain $repr(\mathbf{e})$, that points to a descendent
- Every node ***g*** has an ancestor ***a*** such that $repr(\mathbf{a})$ is the most recently pos-updated descendent of ***g***

Find most recent match of pattern P

- Find point of P on edge $\mathbf{g}$ in suffix tree
- Track back to root via suffix links
= traverse the upward path to root in link tree
- For each $\mathbf{e}$ on the path, if $\mathbf{f}=\text{repr}(\mathbf{e})$ is descendent of $\mathbf{g}$, and $\mathbf{f}$ is more recently updated than any previous encountered, then keep $\mathbf{f}$'s position

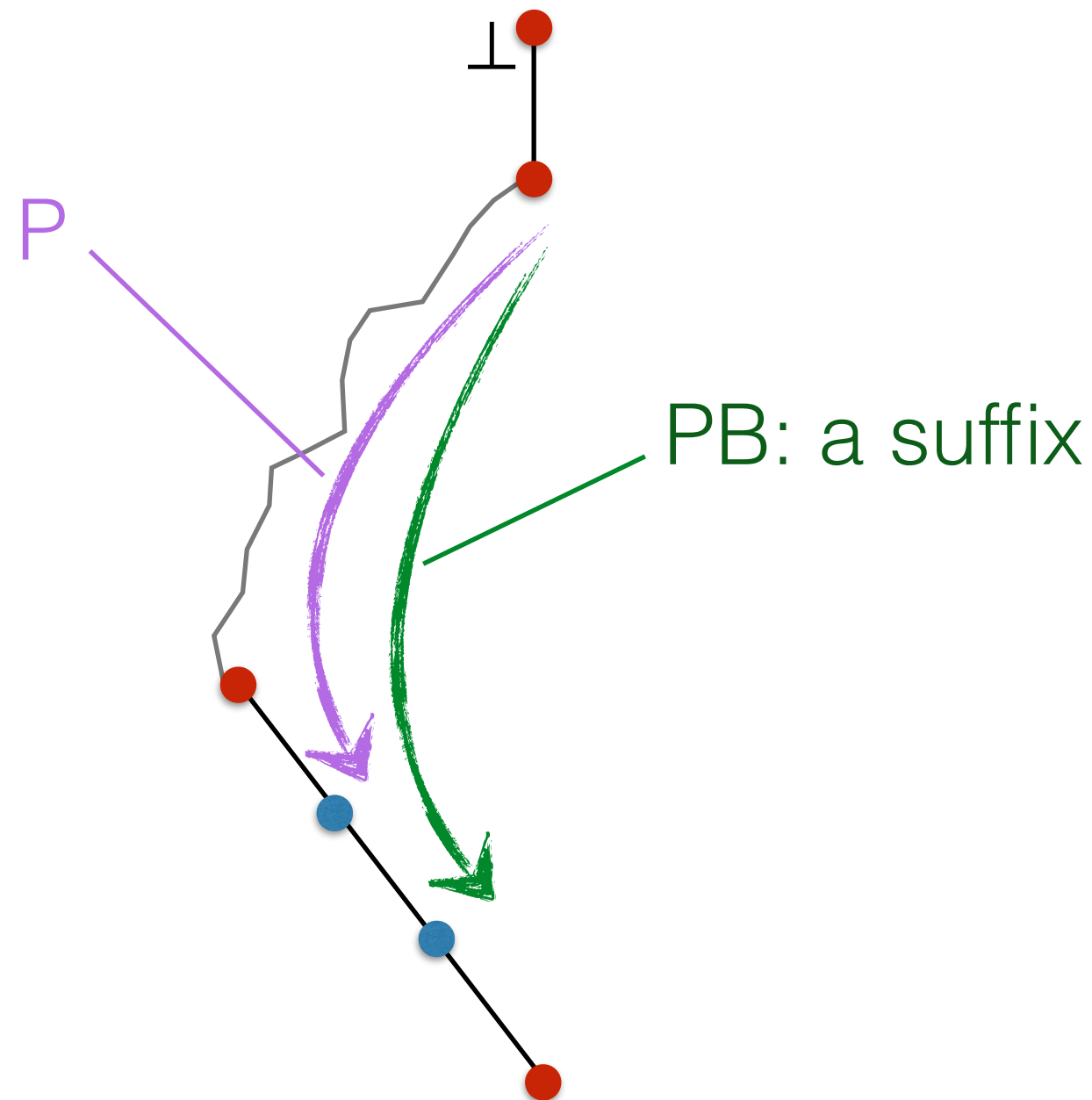
Maintaining property 1

- When pos-updating $\mathbf{e}$, $repr(\vdash) := \mathbf{e}$
- Recursively push the *previous* (overwritten) *repr* value $\mathbf{f}$ down the tree to just below $LCA(\mathbf{e}, \mathbf{f})$



- Worst case: # pushes = height
- Lemma: # pushes in N iterations is $O(N \log N)$
- Proof: $\exists$ corresponding balanced binary tree with $O(N)$ nodes, where # pushes would be at least as many as in the link tree
- (Adversarial string exists, which produces worst case)

Complication: active point is below P on the same edge,
⇒ pos-update is pending



Solution from Breslauer & Italiano 2012: maintain implicit suffix nodes, detect and handle this case

Sliding window of size $W \leq N$

- All data structures can be deconstructed from behind in amortized constant time per iteration (using Fiala & Greene 1989 for suffix tree)
- Main result: MRM queries for pattern P in time $O(|P|)$ maintained in time $O(N \log W)$

Optimization for LZ factorization

- Delay pos-updates until a whole factor is added to tree, and then execute backwards
- Can skip pos-updates that would be superseded by later ones in the same factor
- **Effects:**
Known adversarial string is cyclic: $O(N)$
New worst case: ?

Conclusion

- Showed data structure for general optimal MRM queries can be maintained in $O(N \log N)$ time ($O(N \log W)$ for sliding window size W)

Some interesting remaining questions:

- Is there an $o(N \log N)$ algorithm?
- More practical data structure