# Trie Trees

Ioannis Lamprou

SAV '15, EPFL

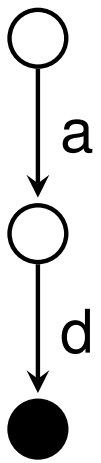
# Trie Trees

## Some Intro

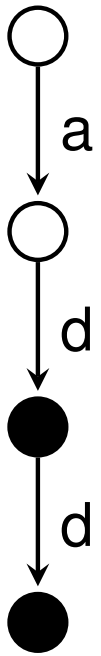
- Re**trie**val trees
- Prefix trees (node $\sim$ prefix)
  - common parent $\rightarrow$ common prefix
- Strings as keys
- Use: dictionaries

# Empty Trie

# Insert ad

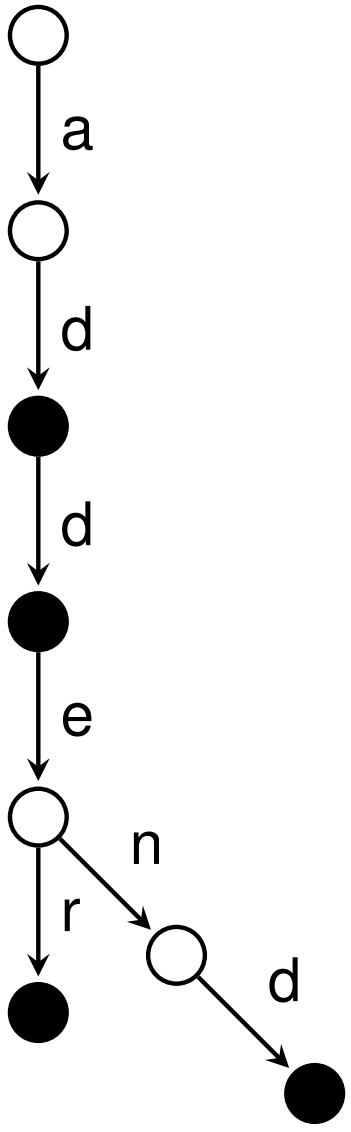


# Insert `add`

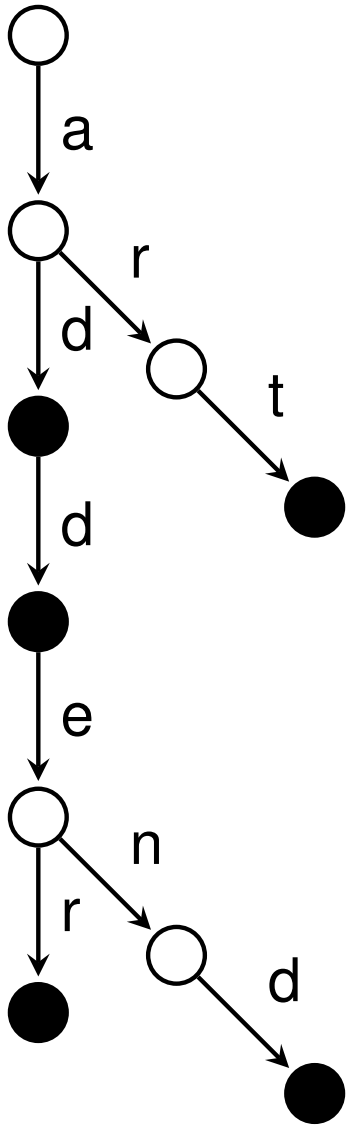


# Insert `adder`

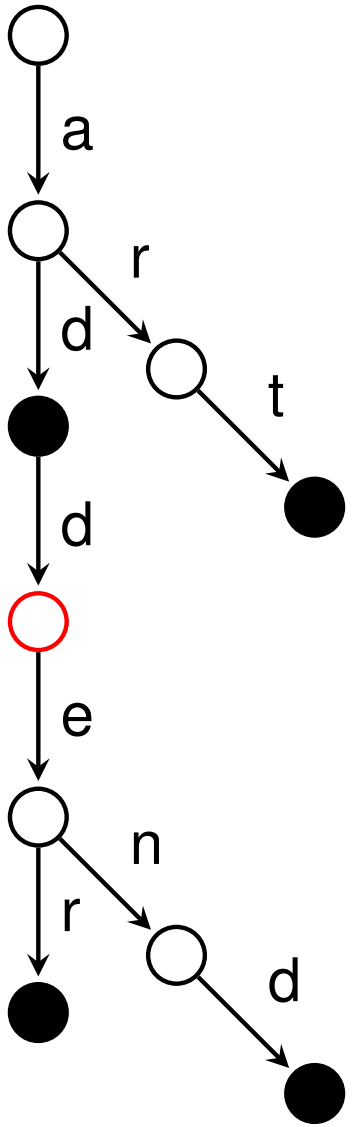
# Insert `addend`



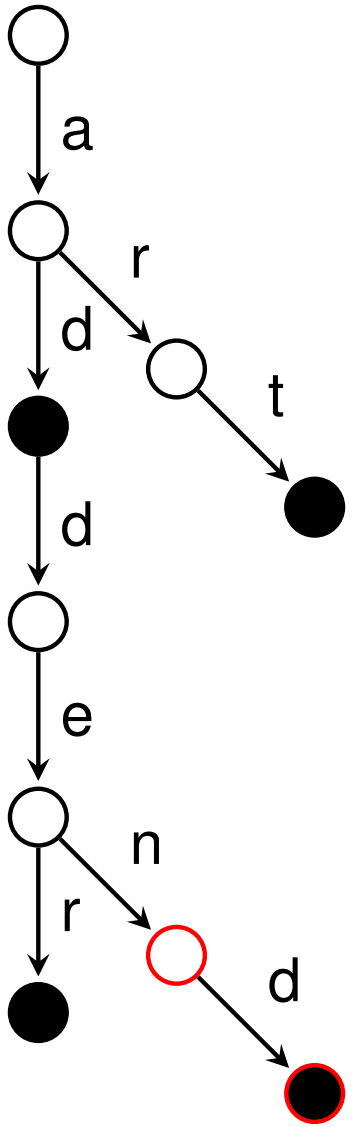
# Insert `art`



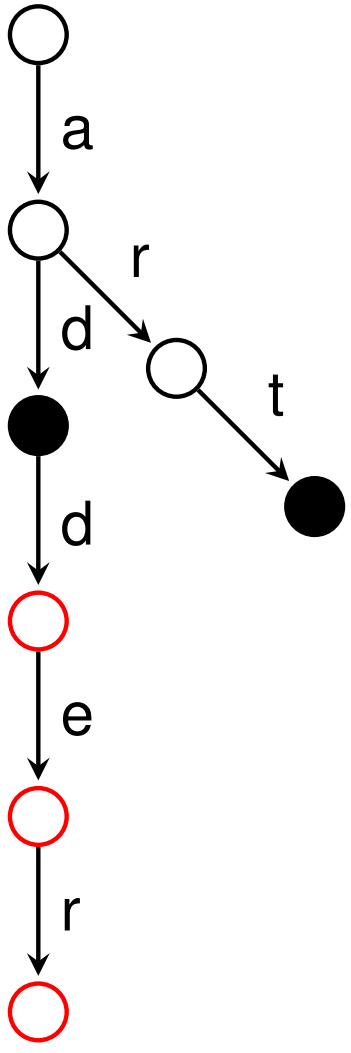
# Delete `add`



# Delete addend



# Delete `adder`



# Conclusions

## So far

- Insertion, Deletion, Find
- Node: Optional Value & List of (Char, Child)
- All leafs are words property
- Verification over strings is hard
- $\rightarrow$ special cases, different representation

## Extra

- Radix trees: compact tries
- Compress nodes with single child
- Edges labeled with strings, not single characters