Iterators for the trie data structure

Kasper Egdø

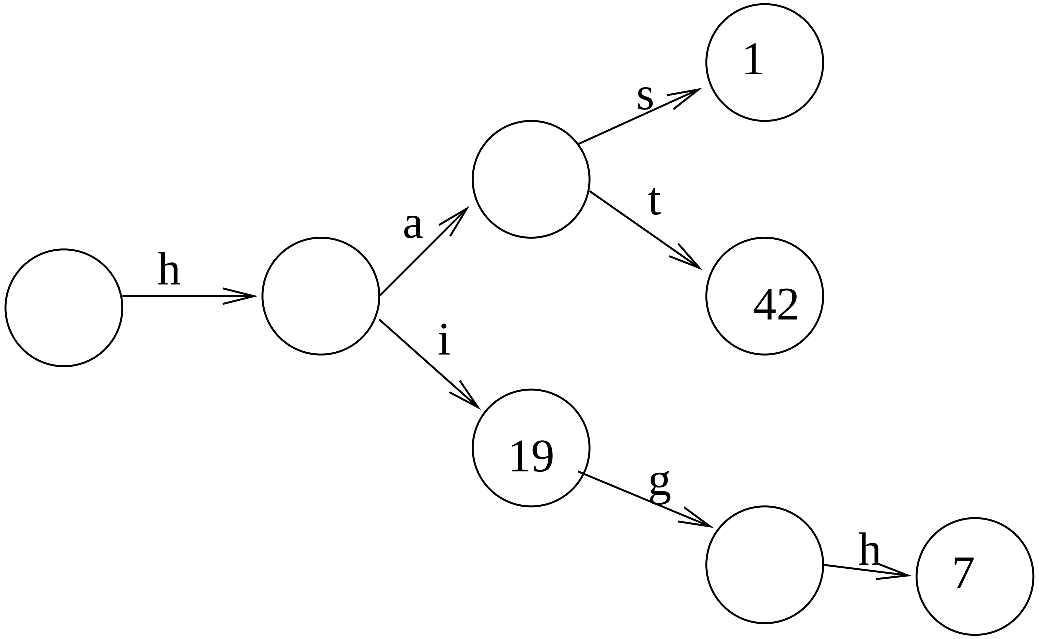
Department of Computer Science

University of Copenhagen

Overview

- Overview of the trie
- Goals: iterators and satellite data for the DAG-representation
- Different representations providing iterators
- The DAG-representation
- Experimental results
- Conclusions

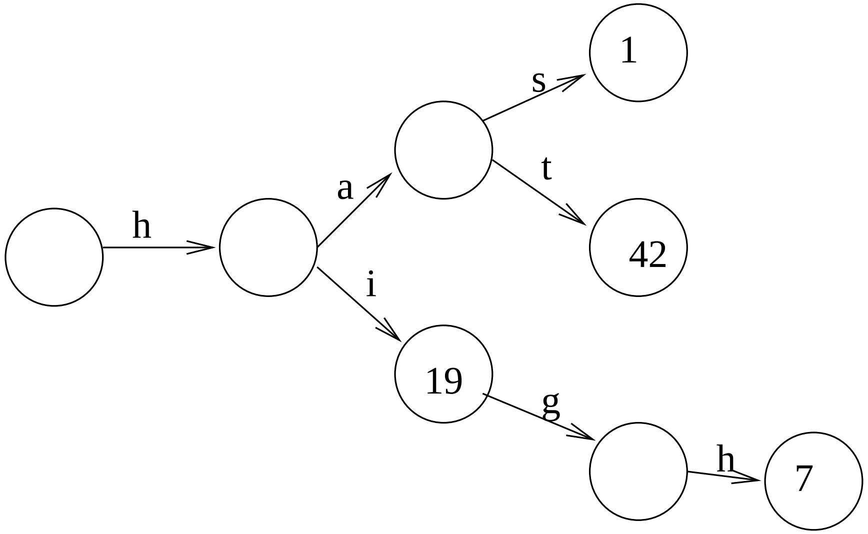
The trie



Storing entries:

- `(has,1)`
- `(hat,42)`
- `(hi,19)`
- `(high,7)`

Operations



- `insert(key, value)`
- `delete(key)`
- `value find(key)`
- `node prefix-match(key)`
- `child-nodes list-children(node)`

Adding entry-iterators

Operations on the data structure become:

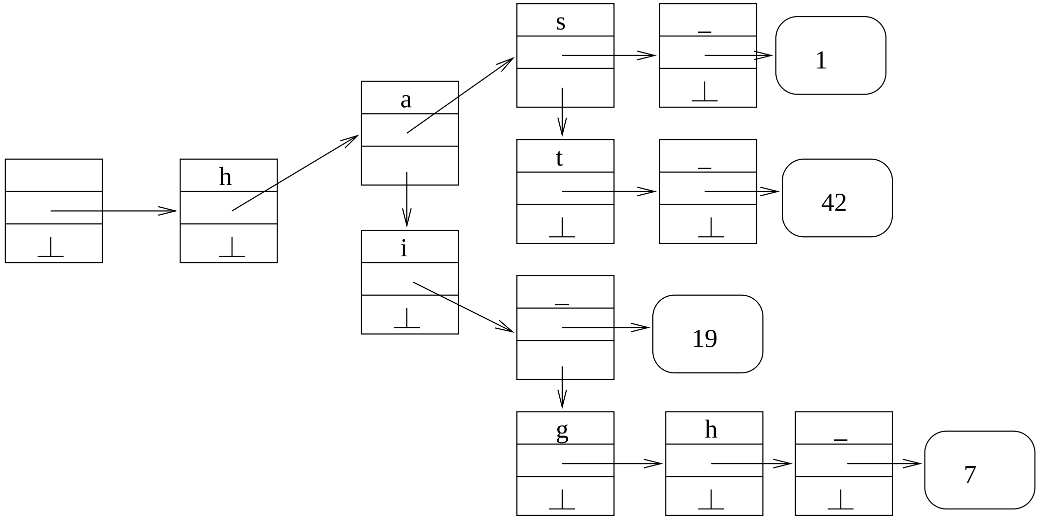
- `iterator insert(key,value)`
- `iterator delete(iterator)`
- `iterator find(key)`
- `node-iterator prefix-match(key)`
- `(node-iterator, node-iterator) list-children(node)`

Operations on iterators:

- `iterator increment()`
- `iterator decrement()`
- `value& dereference()`
- `key getkey()`

Plus the usual suspects: constructors, assignments, equality comparison

The trie represented with linked lists



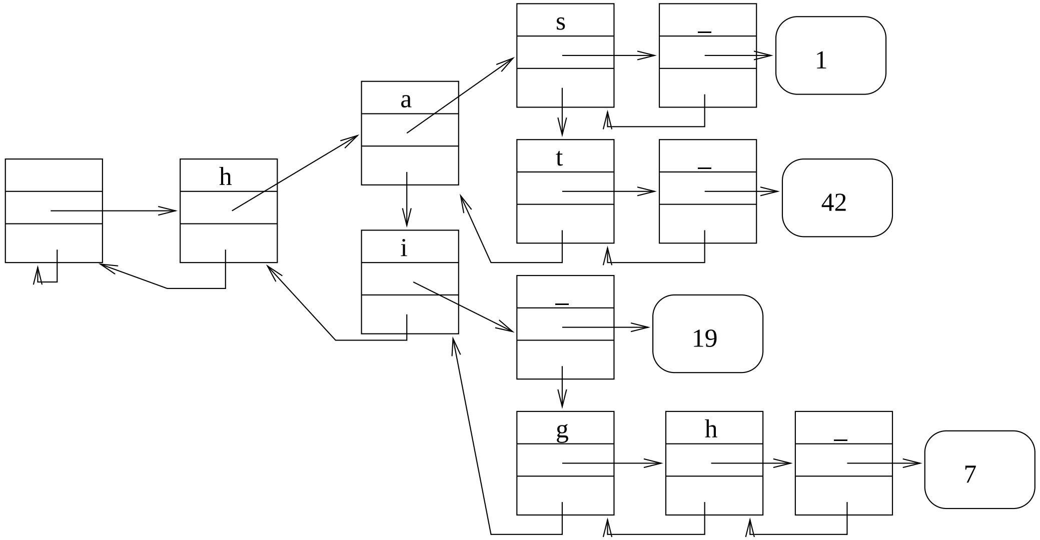
Iterators are pointers to leaf nodes

- `insert` and `delete`: $O(|A|n)$

Adding parent pointers and predecessor/successor pointers (not shown) enables:

- `incrementing iterator`: $O(1)$
- `decrementing iterator`: $O(1)$
- `getkey`: $O(n)$
- `dereferencing iterator`: $O(1)$

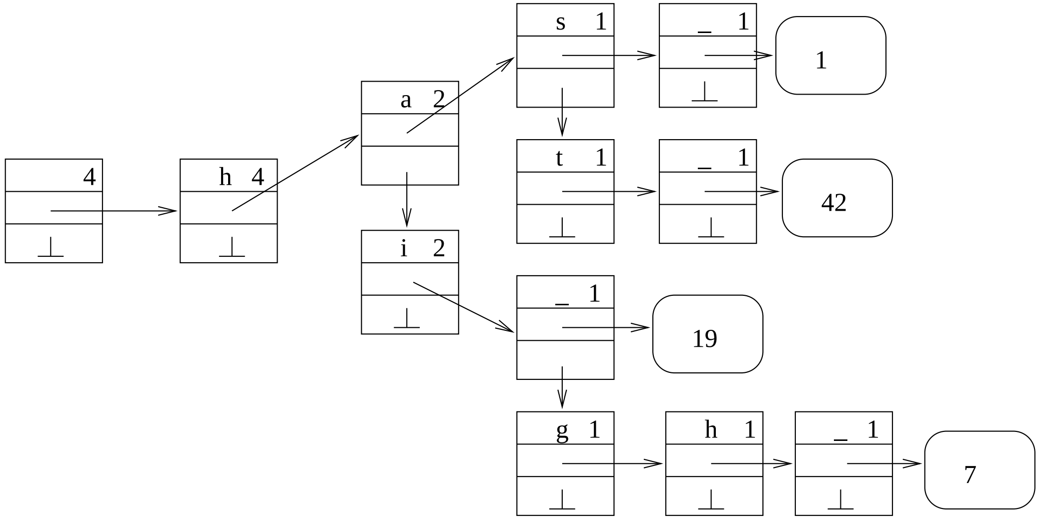
The trie with parent links



Iterators are pointers to leaf nodes. Uses no extra pointers.

- `insert` and `delete`: $O(|A|n)$
- `incrementing iterator`: $O(|A|n)$
- `decrementing iterator`: $O(|A|^2n)$
- `getkey`: $O(|A|^2n)$
- `dereferencing iterator`: $O(1)$

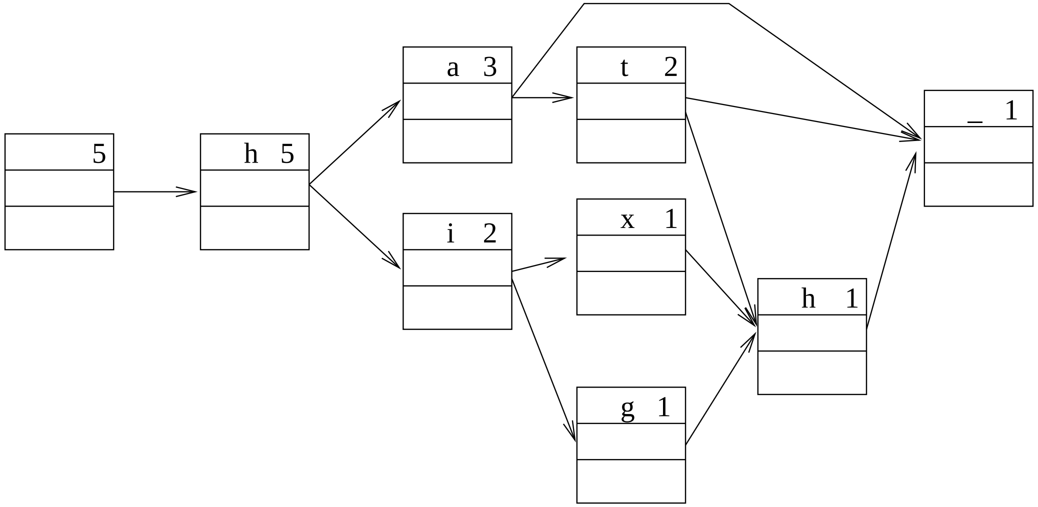
The trie with counting



Iterators are integers. Uses an integer per node.

- insert and delete: $O(|A|n)$
- incrementing iterator: $O(1)$
- decrementing iterator: $O(1)$
- getkey: $O(|A|n)$
- dereferencing iterator: $O(|A|n)$

The DAG-representation



Iterators are integers.

- `insert` and `delete`: Not possible
- `incrementing iterator`: $O(1)$
- `decrementing iterator`: $O(1)$
- `getkey`: $O(|A|n)$
- `dereferencing iterator`: $O(|A|n)$

Experiment

- WordNet nouns
  - 268.984 entries
  - $2.636.945$ letters, not counting terminators
- Trie
  - $1.333.594$ nodes
- DAG
  - 270.488 nodes
  - 1.85 average fanout

Conclusions

- Old but relevant
- Iterators can be a free wrt space
- The DAG can have satellite data
- The DAG representation can be effective