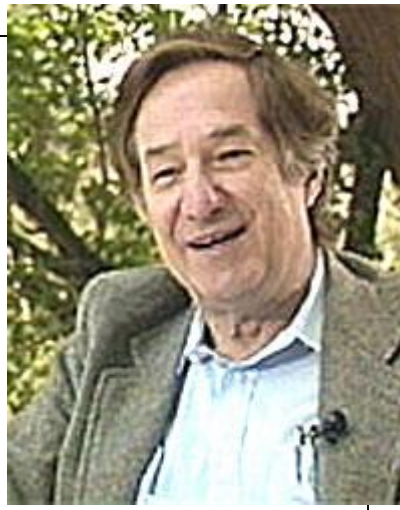
Topic 24
Tries



"In 1959, (Edward) Fredkin recommended that BBN (Bolt, Beranek and Newman, now BBN Technologies) purchase the very first PDP-1 to support research projects at BBN. ***The PDP-1 came with no software whatsoever.***

Fredkin wrote a PDP-1 assembler called FRAP (Free of Rules Assembly Program);"

Tries were first described by René de la Briandais in *File searching using variable length keys*.

Clicker 1

- How would you pronounce “Trie”

A. “tree”

B. “tri – ee”

C. “try”

D. “tiara”

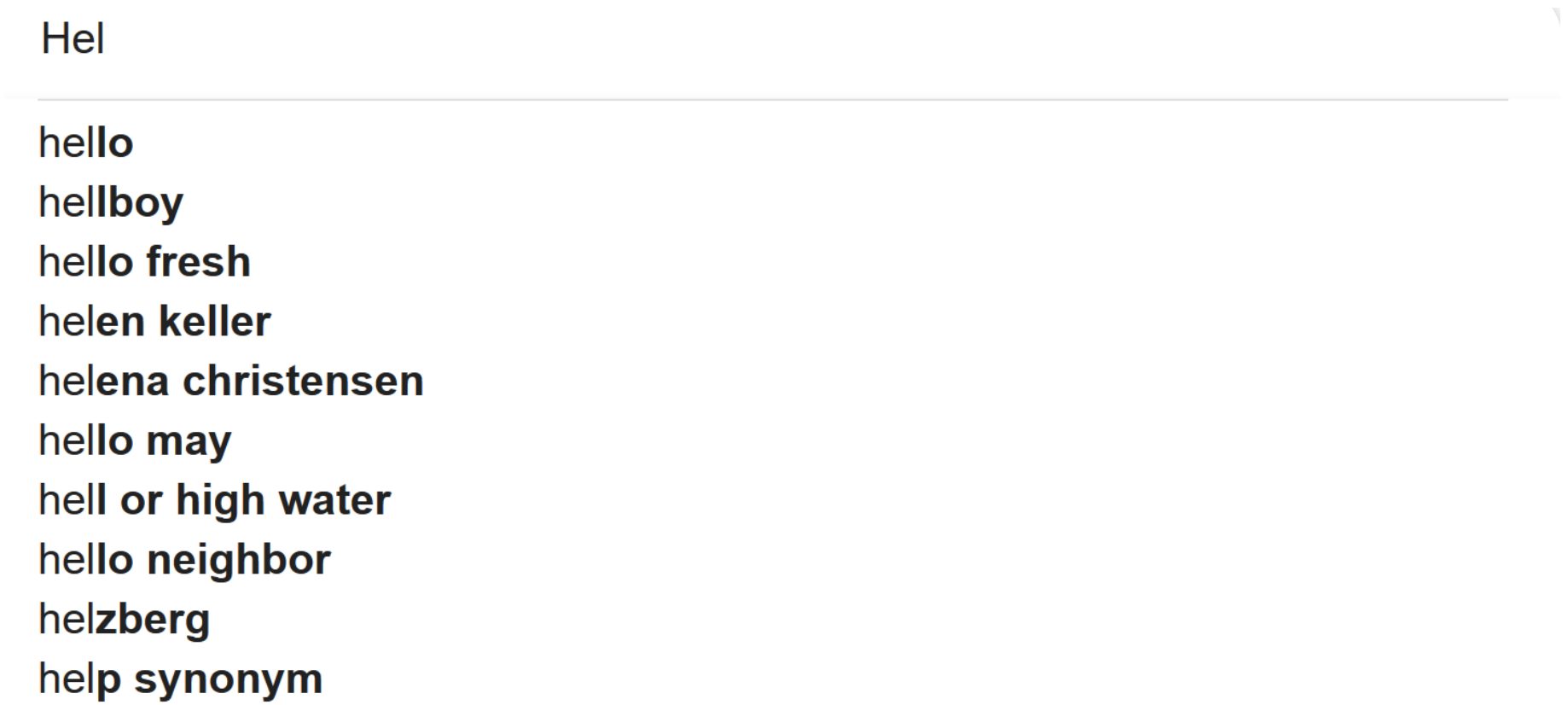
E. something else

Tries aka Prefix Trees

- Pronunciation:
- From re**trie**val
- Name coined by Computer Scientist Edward Fredkin
- Re**trie**val so “tree”
- … but that is very confusing so most people pronounce it “try”

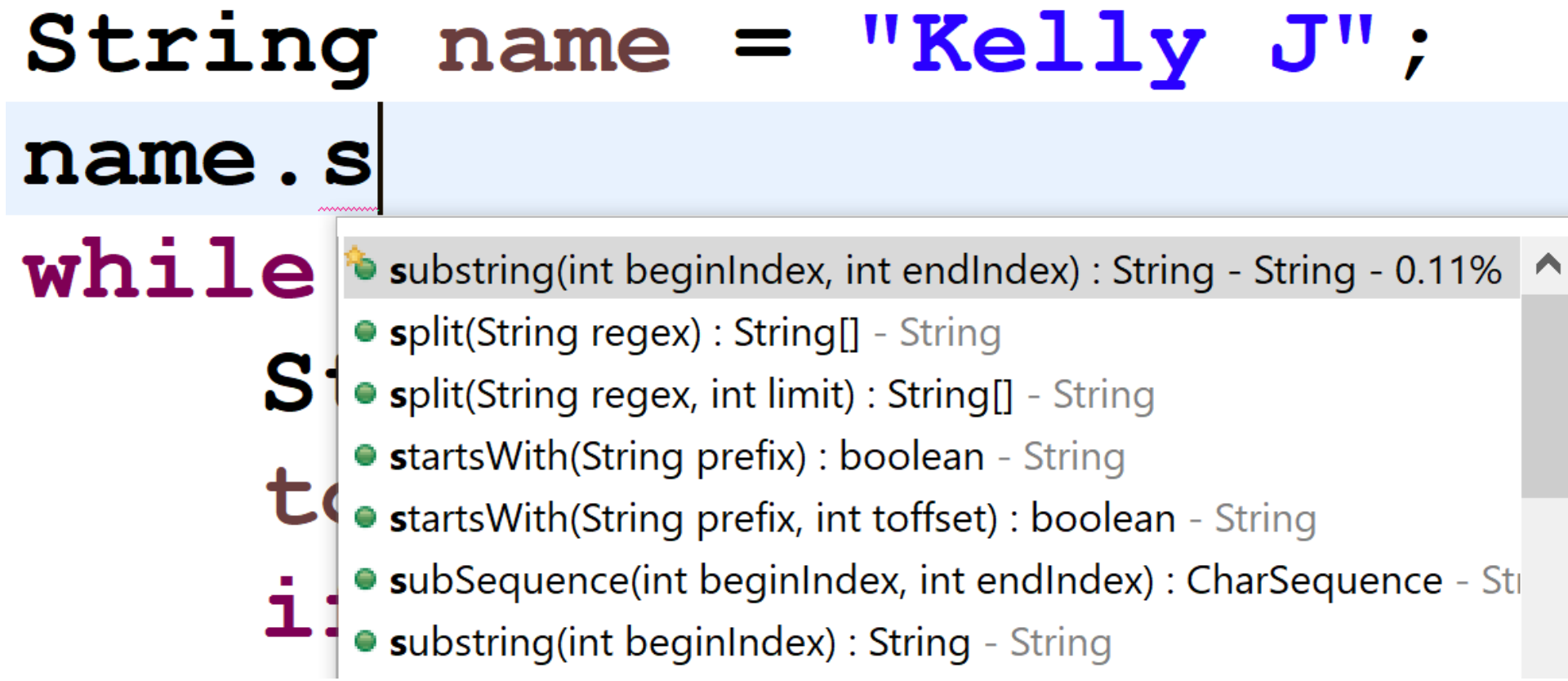
Predictive Text and AutoComplete

- Search engines and texting applications guess what you want after typing only a few characters



AutoComplete

- So do other programs such as IDEs



Searching a Dictionary

- How?
- Could search a set for all values that start with the given prefix.
- Naively O(N) (search the whole data structure).
- Could improve if possible to do a binary search for prefix and then localize search to that location.

Tries

- A general tree (more than 2 children possible)
- Root node (or possibly a list of root nodes)
- Nodes can have many children
  - not a binary tree
- In simplest form each node stores a character and a data structure (list?) to refer to its children
- "Stores" all the words or phrases in a dictionary.
- How?

René de la Briandais Original Paper

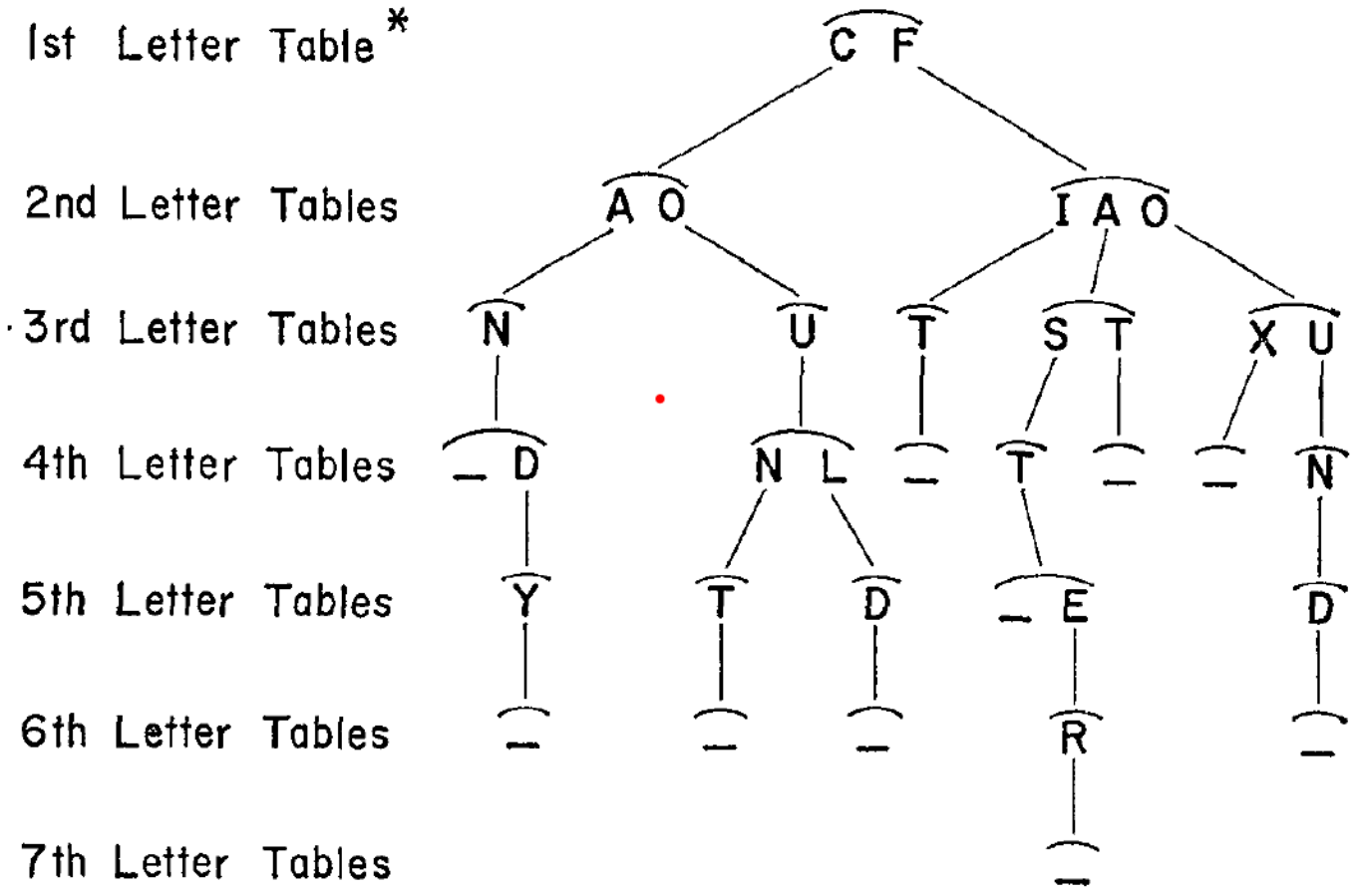
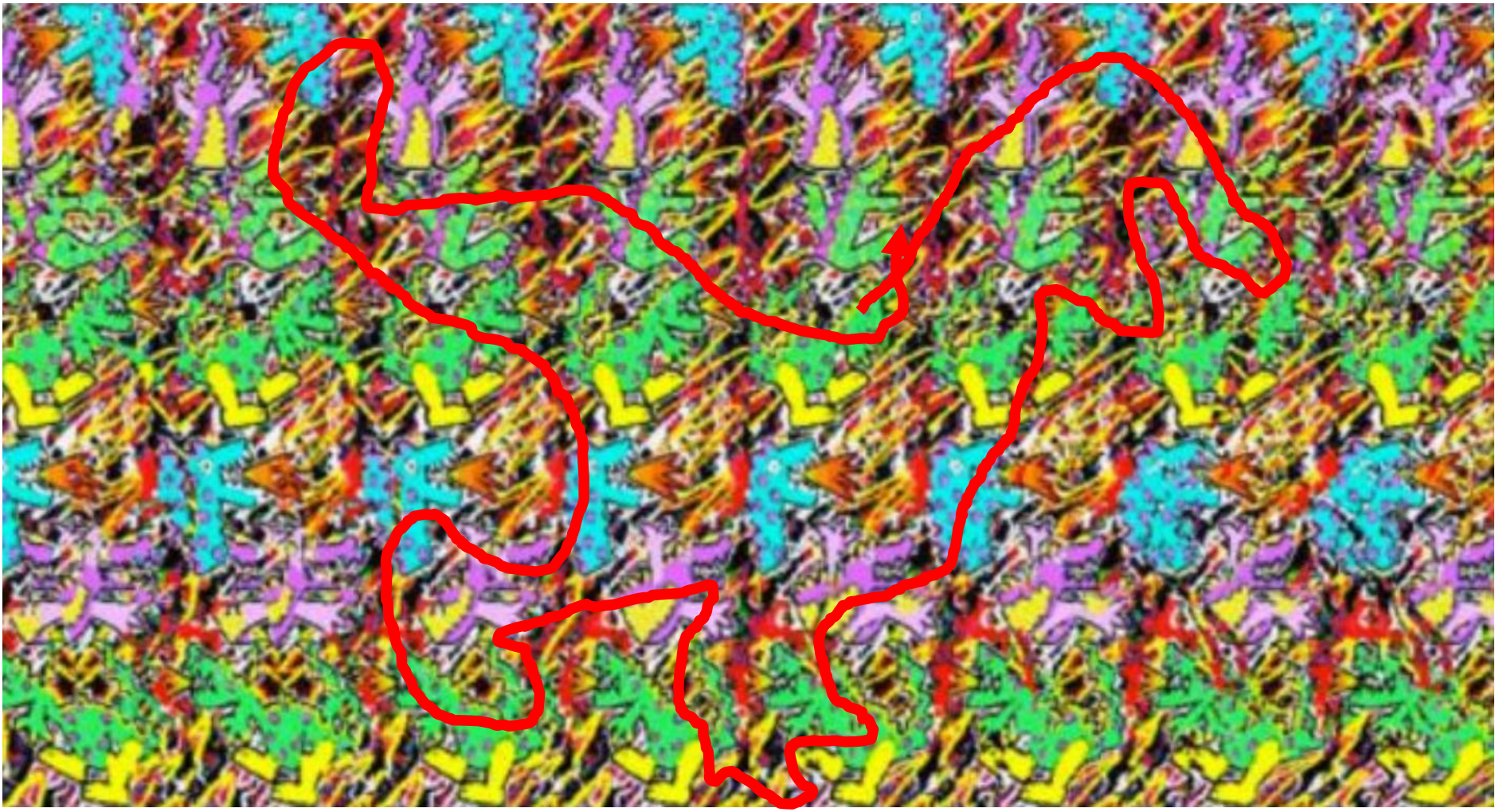


Fig. 1—Formation of a set of tables.

????

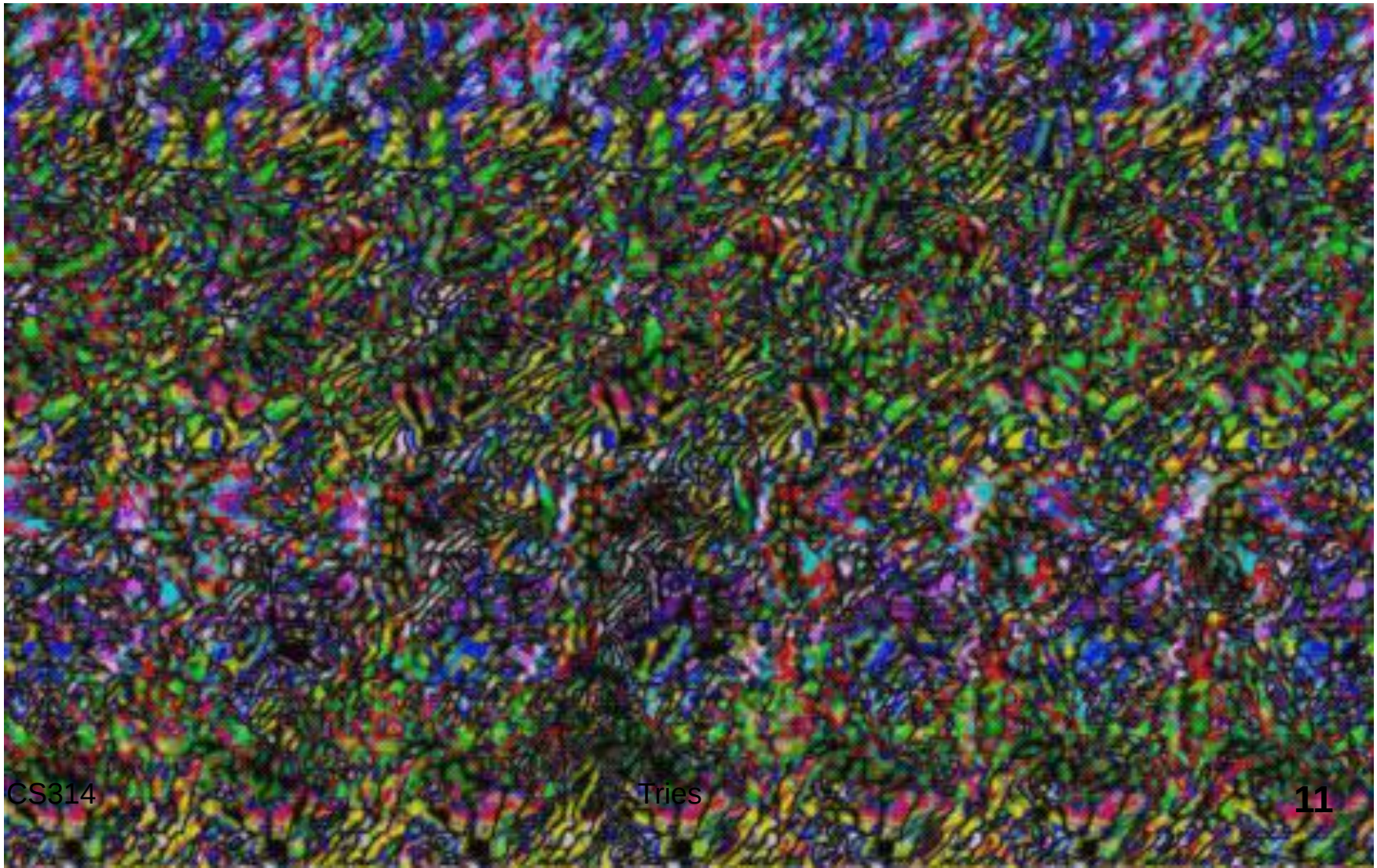
????



Picture of a Dinosaur

Fall 2022 - Ryan P.

Created with Procreate: https://procreate.art/



Can

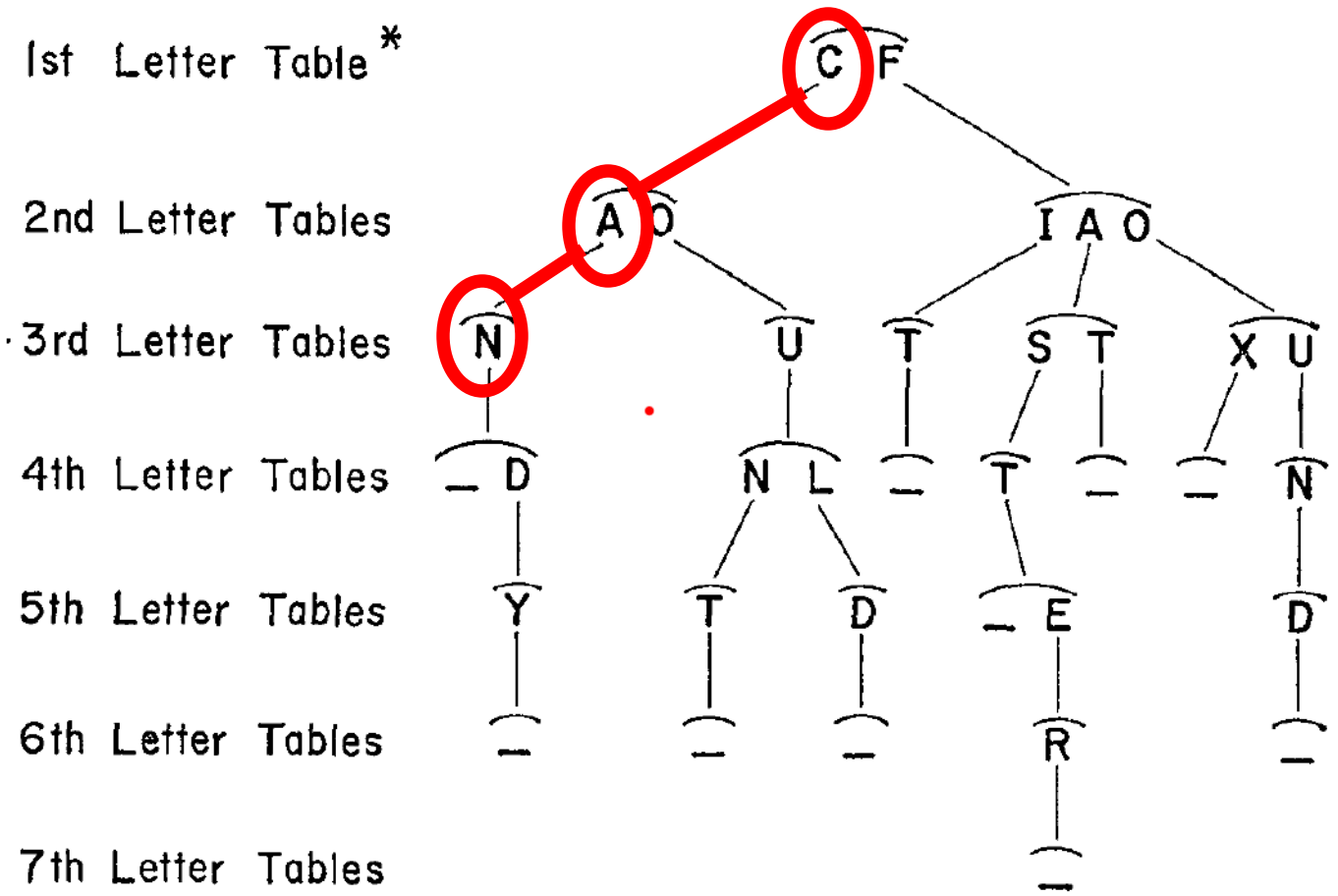


Fig. 1—Formation of a set of tables.

Candy

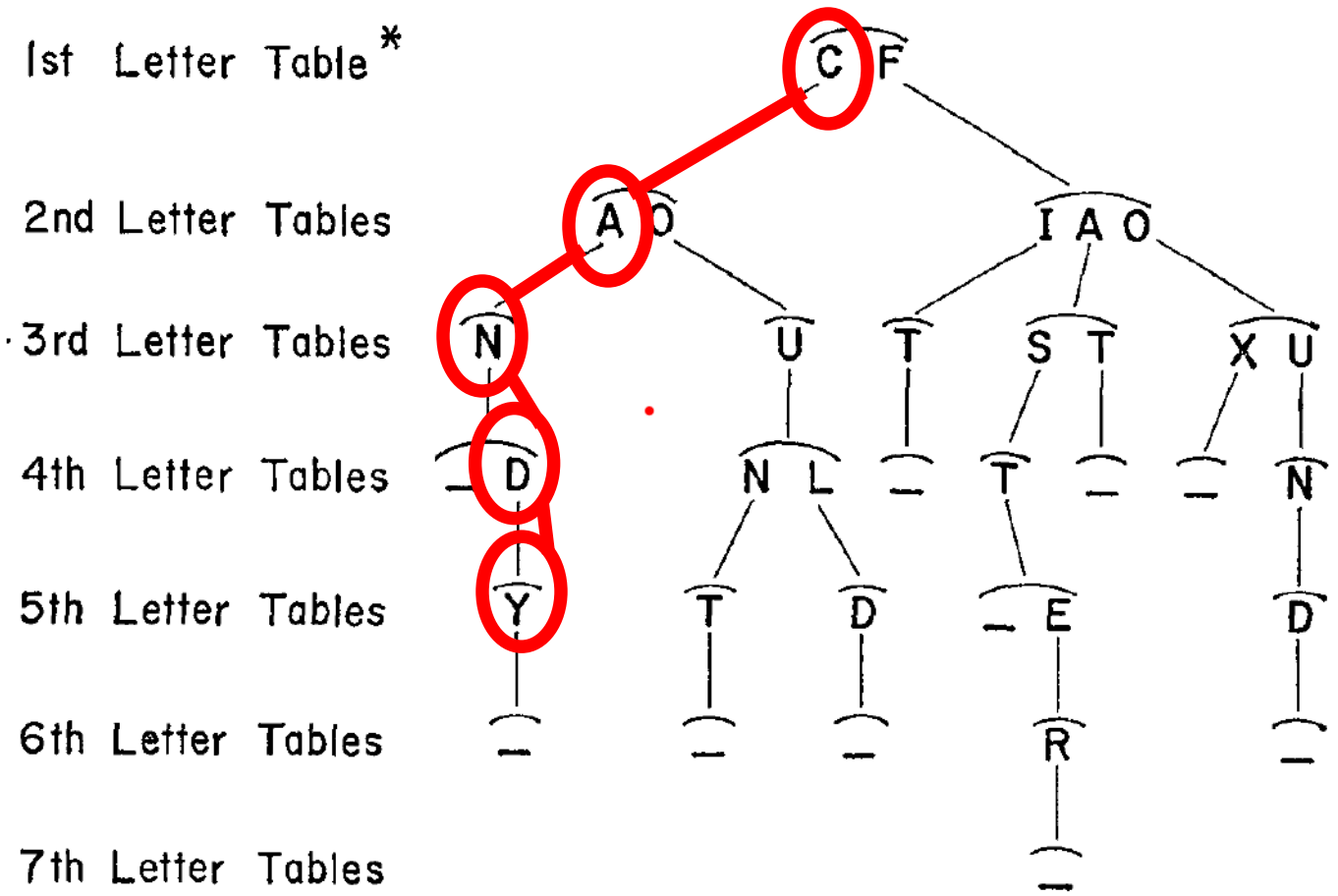


Fig. 1—Formation of a set of tables.

Fox

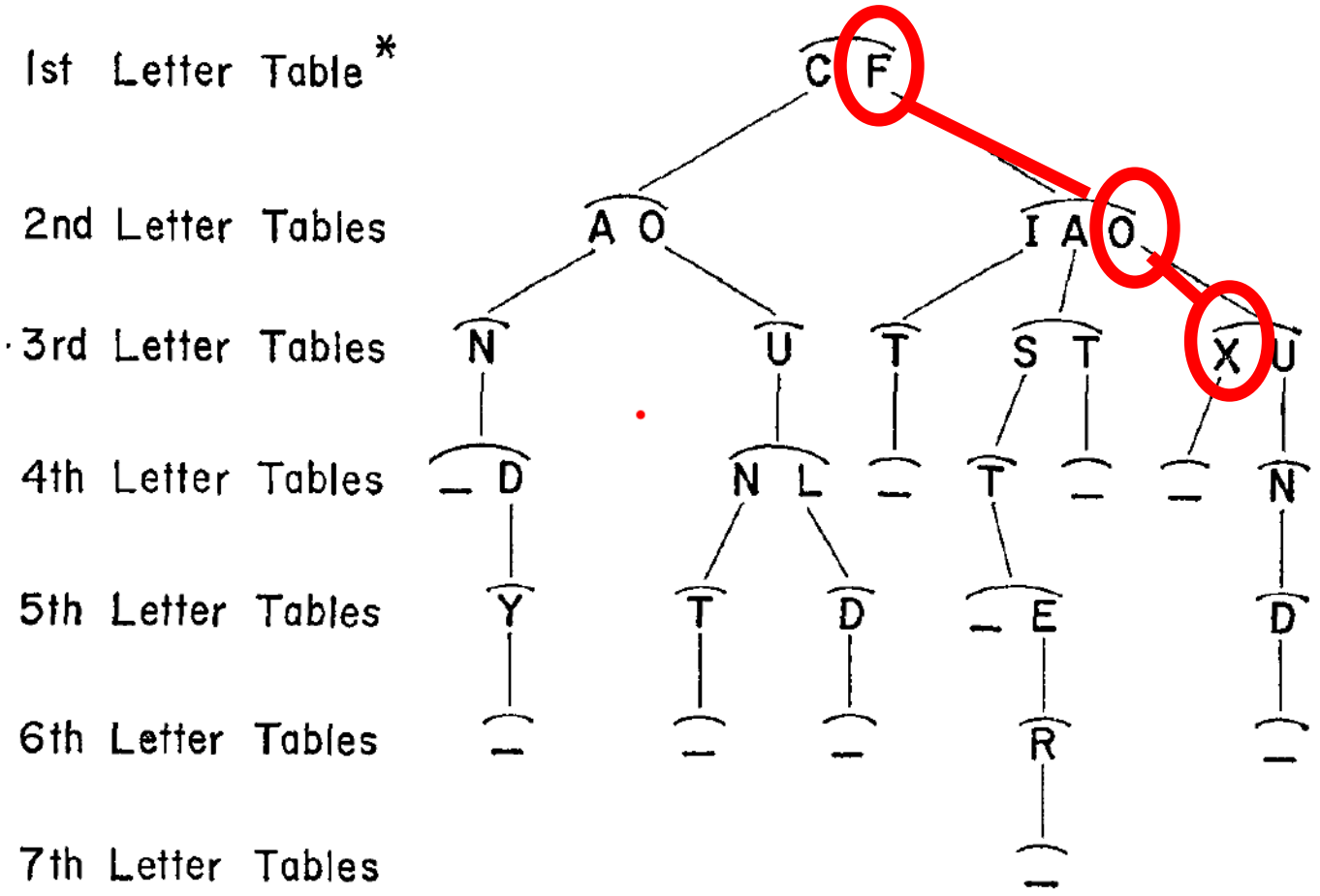


Fig. 1—Formation of a set of tables.

Clicker 2

- Is “fast” in the dictionary represented by this Trie?

A. No

B. Yes

C. It depends

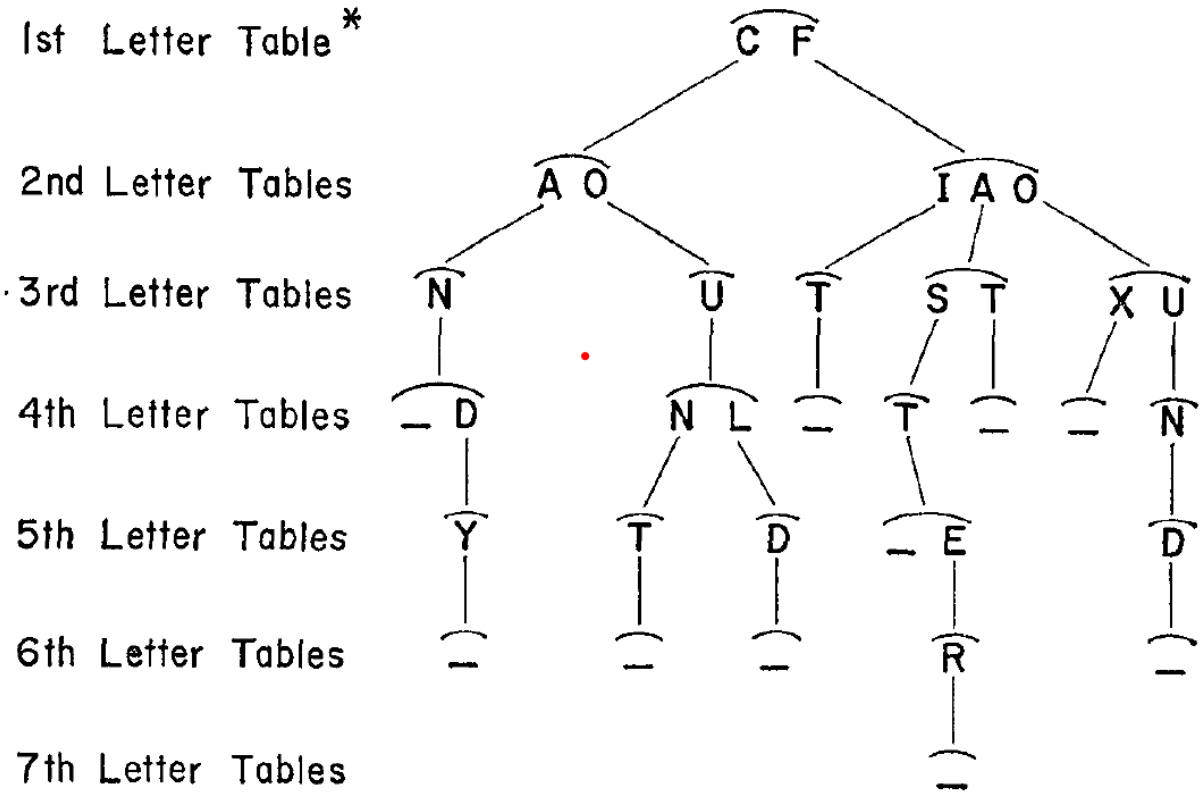


Fig. 1—Formation of a set of tables.

Clicker 3

- Is “fist” in the dictionary represented by this Trie?

A. No

B. Yes

C. It depends

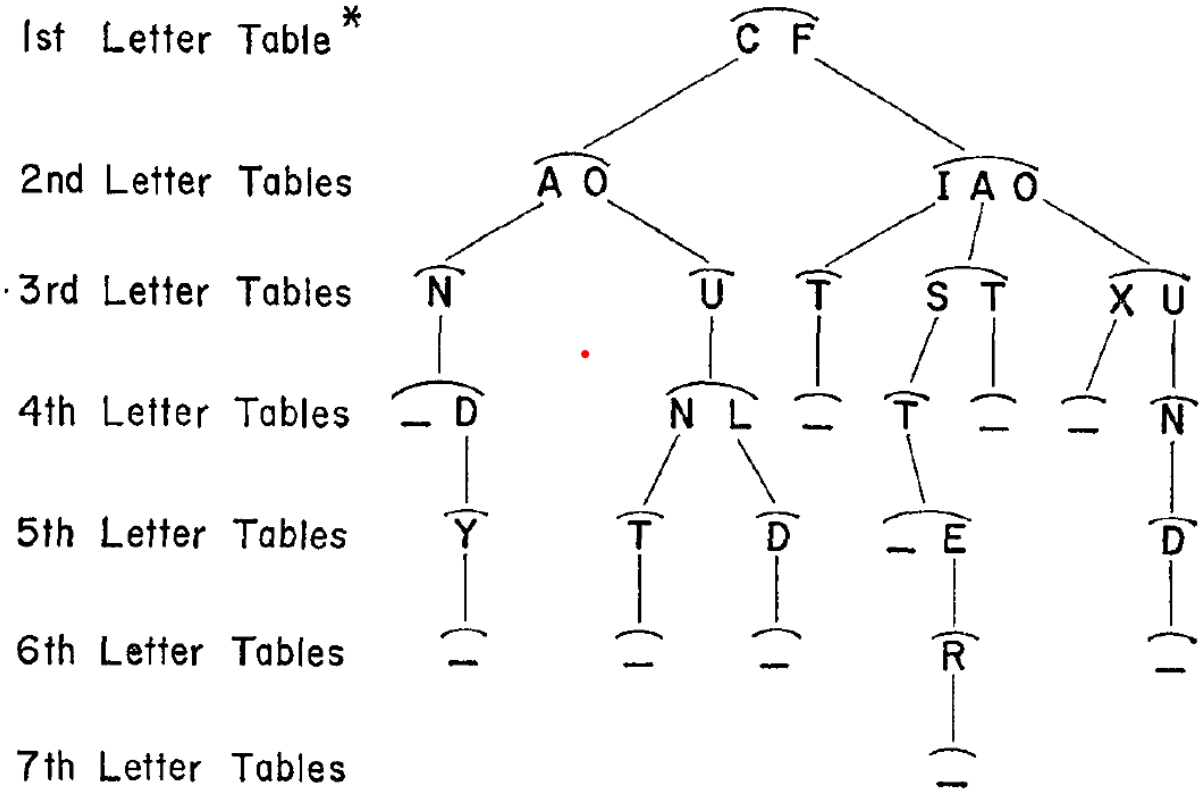
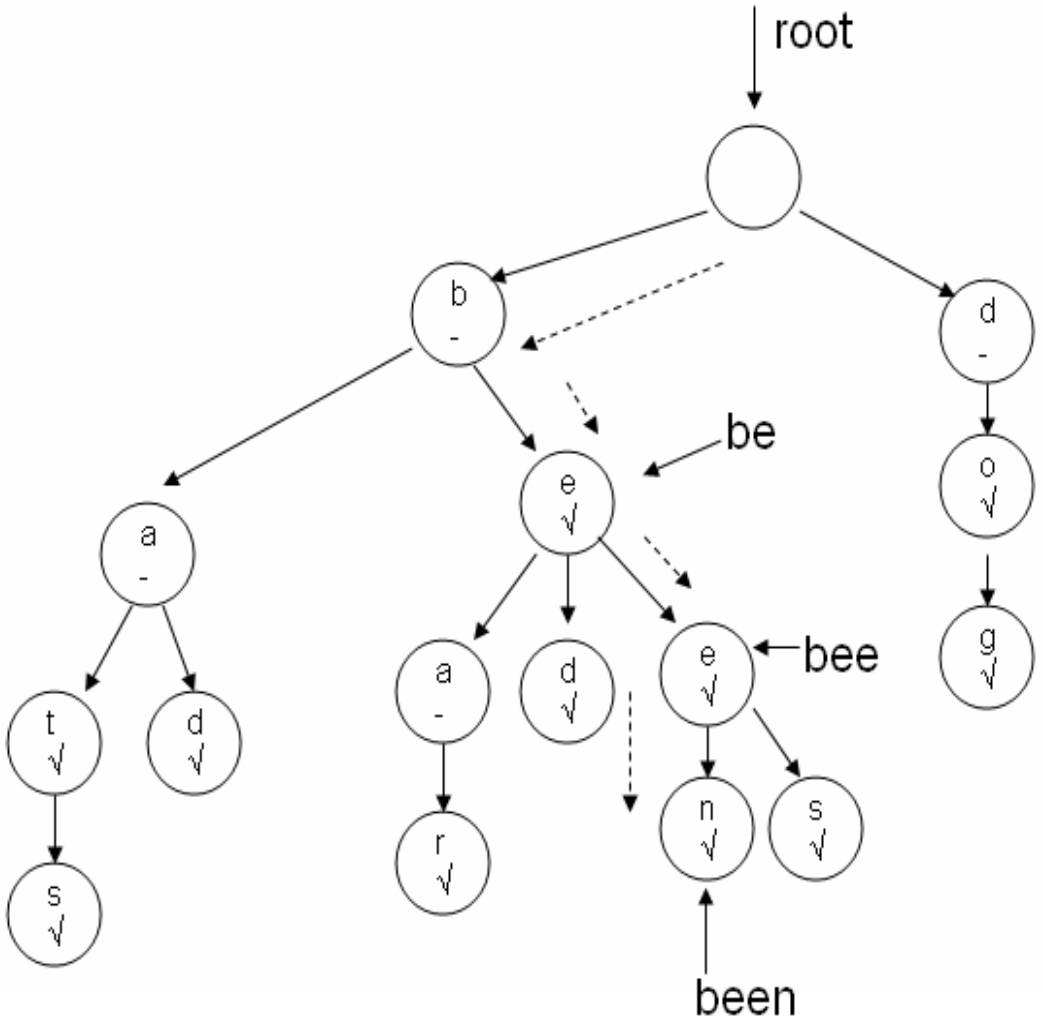


Fig. 1—Formation of a set of tables.

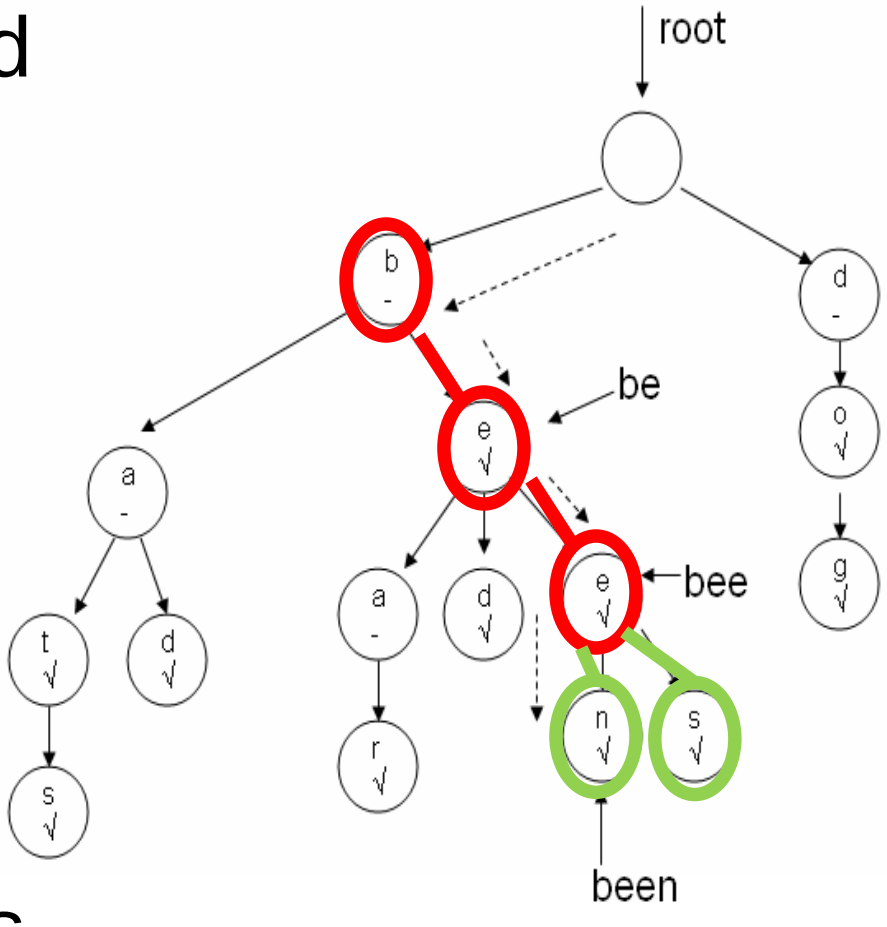
Tries

- Another example of a Trie
- Each node stores:
  - A char
  - A boolean indicating if the string ending at that node is a word
  - A list of children



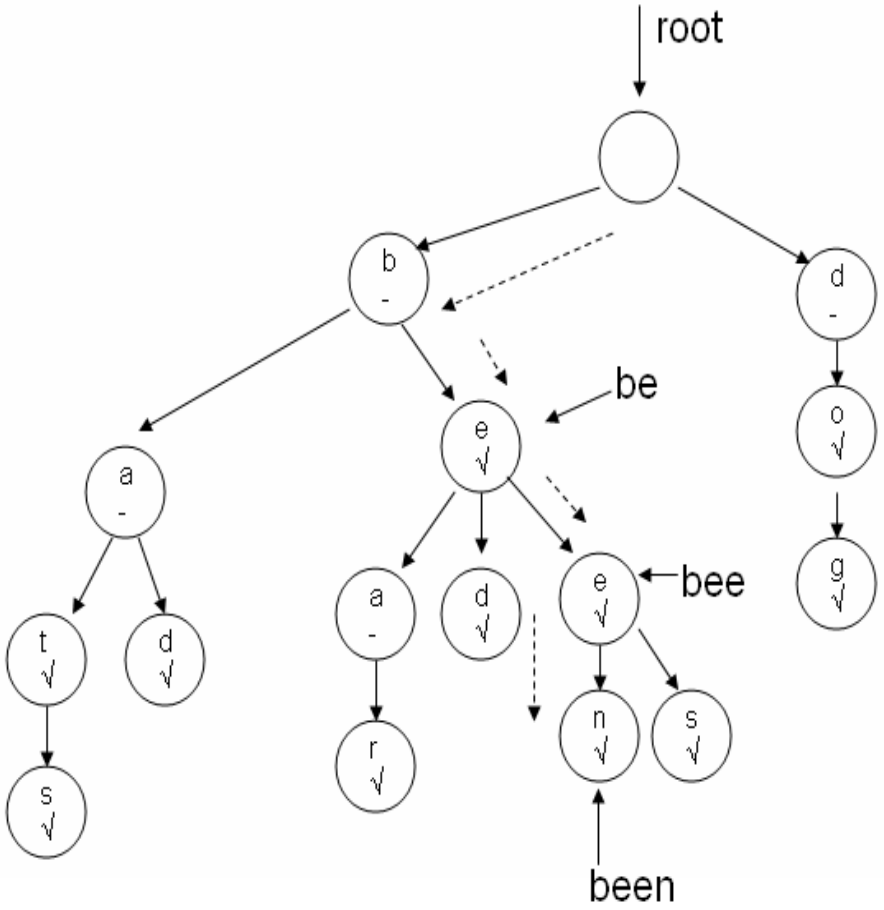
Predictive Text and AutoComplete

- As characters are entered we descend the Trie
- … and from the current node …
- … we can descend to terminators and leaves to see all possible words based on current prefix
- b, e, e -> bee, been, bees



Tries

- Stores words and phrases.
  - other values possible, but typically Strings
- The whole word or phrase is not actually stored in a single node.
- … rather the path in the tree represents the word.



Implementing a Trie

```
public class Trie {

    private TNode root;
    private int size; // number of words
    private int numNodes;

    public Trie() {
        root = new TNode();
        numNodes = 1;
```

TNode Class

```
private static class TNode {
        private boolean word;
        private char ch;
        private LinkedList<TNode> children;
```

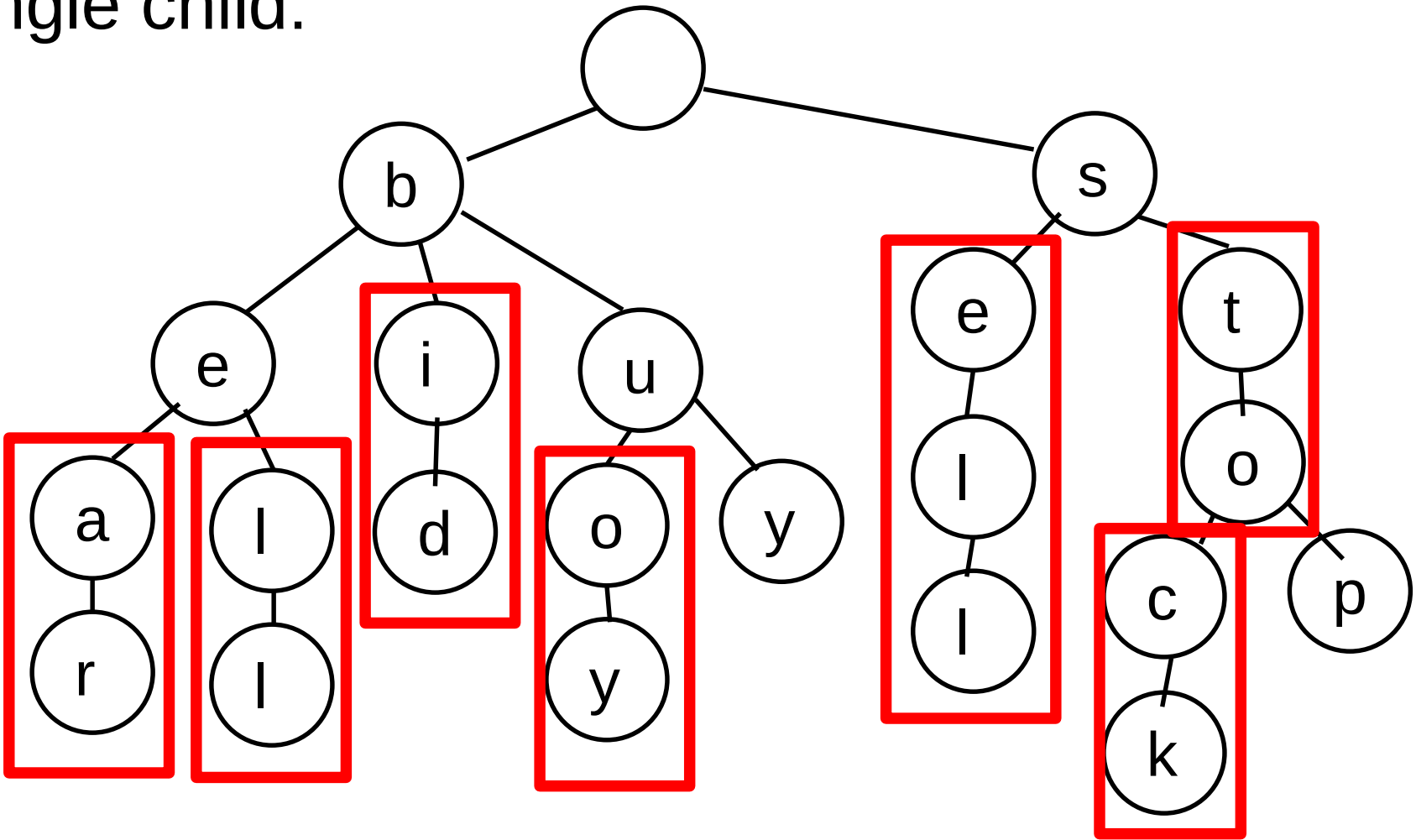
- Basic implementation uses a LinkedList of TNode objects for children
- Other options?
  - ArrayList?
  - Something more exotic?

Basic Operations

- Adding a word to the Trie
- Getting all words with given prefix
- Demo in IDE

Compressed Tries

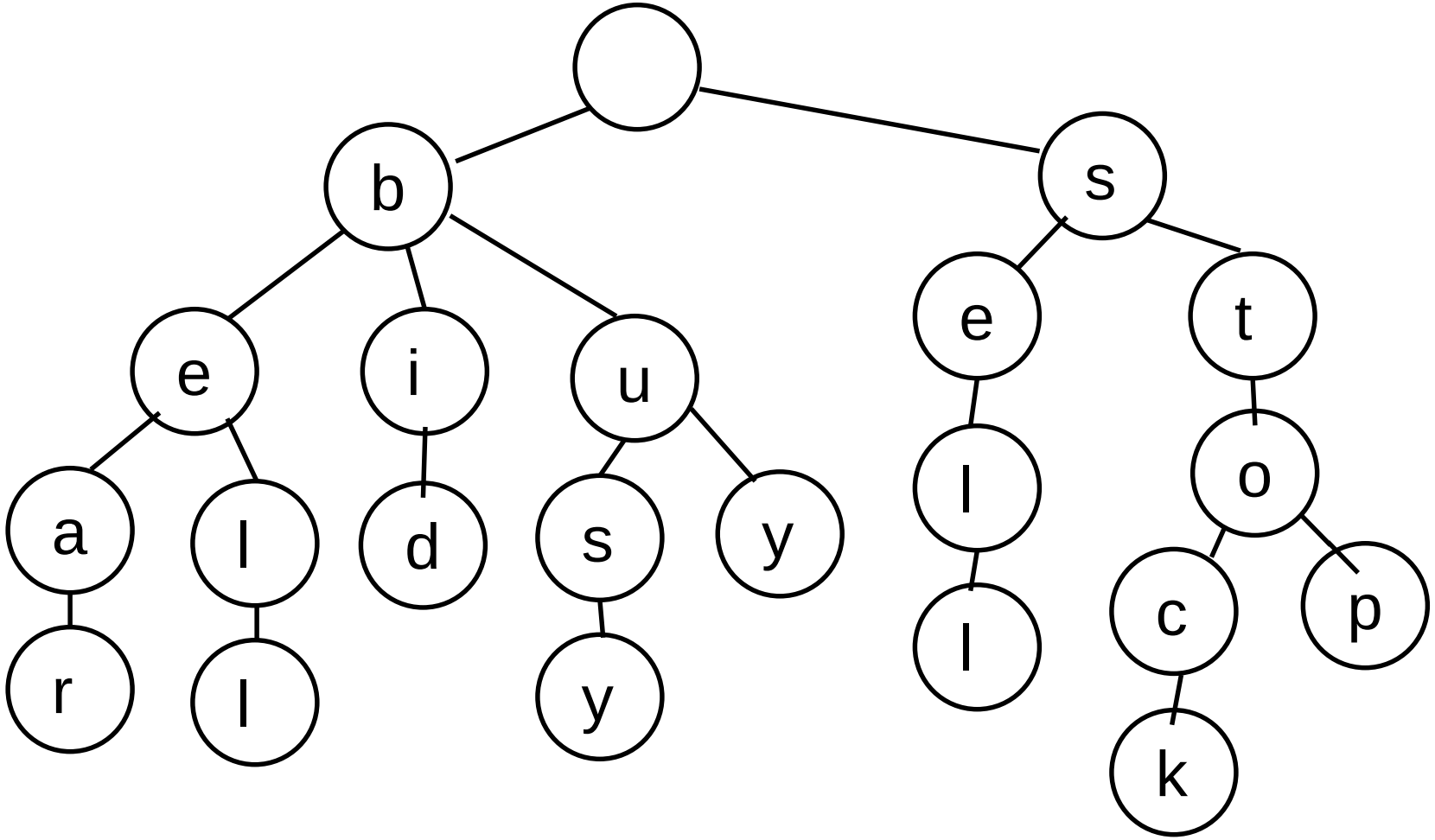
- Some words, especially long ones, lead to a chain of nodes with single child, followed by single child:



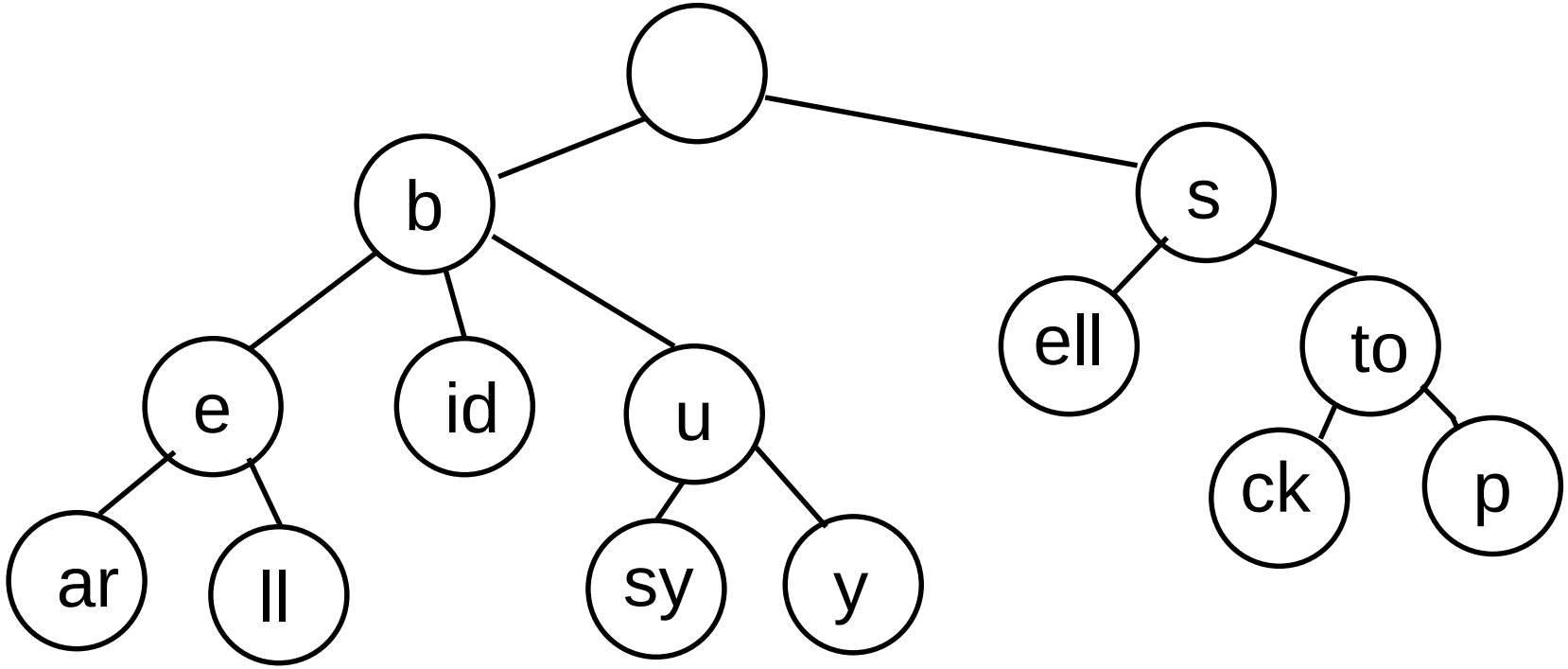
Compressed Trie

- Reduce number of nodes, by having nodes store Strings
- A chain of single child followed by single child (followed by single child …) is compressed to a single node with that String
- Does not have to be a chain that terminates in a leaf node
  - Can be an internal chain of nodes

Original, Uncompressed



Compressed Version



8 fewer nodes compared to uncompressed version

s – t – o – c - k