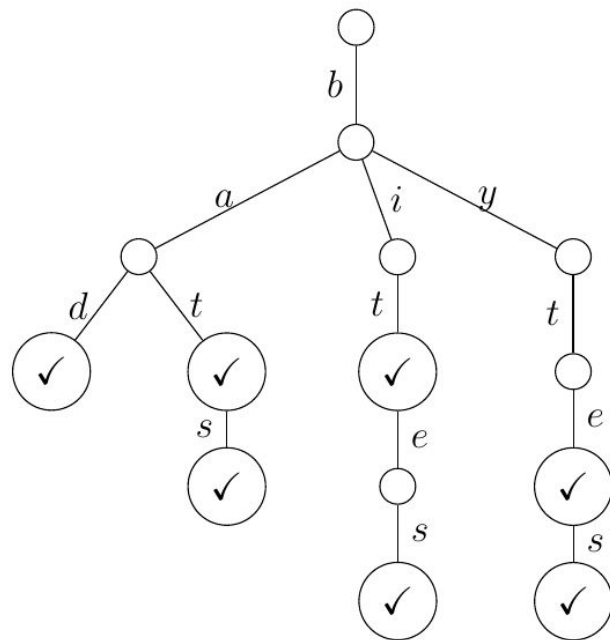


Trie Search

```
contains(T,key):  
  if T is null:  
    return false (for Set) or null (for Map)  
  
  if key is empty string:  
    return T.terminates (for Set) or T.value (for Map)  
  
  return contains(T.children.get(key[0]),rest(key))
```

Examples:

- Search for 'bat'
- Search for 'by'
- Search for 'bag'



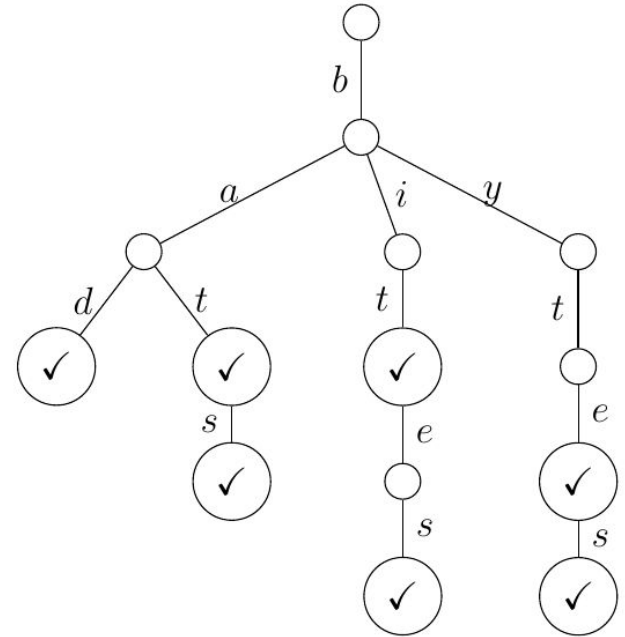
Trie Insert

Basic idea:

- Search along the path specified by key
- Create new TrieNodes along the way if needed
- Upon arriving at the final TrieNode
 - set terminates to true (for Set) or
 - set value to specified value (for Map)

Examples (assuming Set ADT):

- Insert 'bat'
- Insert 'by'
- Insert 'bag'



Trie Delete

Basic idea:

- Search for key
- If not found:
 - Do nothing!
- If found:
 - Set terminates to false (for Set) or value to null (for Map)

Deleting by toggling a flag rather than freeing memory is sometimes called **“lazy deletion.”**

Delete (assuming Set ADT):

- Delete ‘bat’
- Delete ‘bits’
- Delete ‘by’

