

Towards a Formal Verification of B+ Trees

Nicolai Dahl Blicher-Petersen, Christian Harrington, and Morten Fangel Jensen
{ndbl, cnha, mfan}@itu.dk

IT University of Copenhagen, Rued Langgaards Vej 7, 2300 Copenhagen S, Denmark

Abstract. Abstract

Keywords: B+ tree, Coq, Gallina, Formal verification

1 Indexing

- Talk about hyphens and how we should exclude them

Explain your inverted index implementation. As part of your explanation you need to clearly describe how you: • gather term information while your index program is parsing the collection • tokenise terms, including how you deal with punctuation and markup tags, and handle acronyms and hyphenated words • merge the information together for the final lexicon and invlists files This explanation should be around a page in length, but no longer than one and a half pages.

2 Stopping

- Talk about the delegation design pattern we've used? - Talk about the hashing function

Briefly explain how you implemented your stoplist module, including the hash function that you used, and why it is appropriate. This explanation should be around half a page in length.

3 Querying

Explain your index search implementation. As part of your explanation you need to clearly describe: • the data structures you used for your lexicon when it is held in memory • how you look up corresponding term occurrence information in your invlists file This explanation should be around a page in length, but no longer than one and a half pages.

How large is your inverted index? How does this compare with the size of the original collection? Why is it larger/smaller than the original collection?

A Find Subtree Source