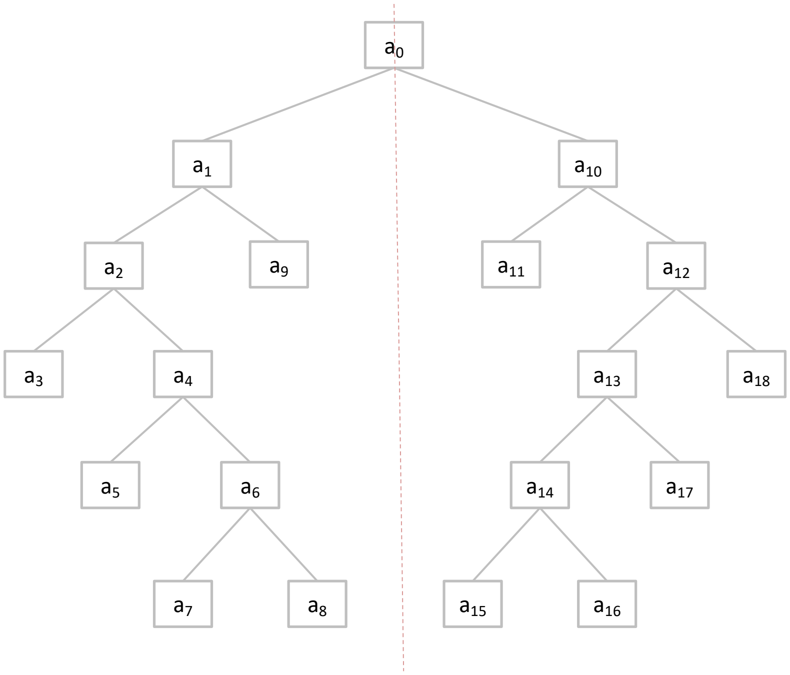
COMP-4400 Winter 2024 Assignment

Logic Programming – Symmetric Binary Trees

A binary tree is either composed of a single node or composed of a root node and exactly two subtrees, which are also binary. A binary tree is symmetric if we can draw a vertical line through the root node so that the subtree on the right-hand side mirrors the one on the left-hand side. Note that this symmetry is defined only on the structure of the tree, not including the content of each node. The following is a symmetric binary tree.



Write prolog code so we can check whether a binary tree is symmetric. You do not need to consider trees that are not binary.

You can use \+. You cannot use predicates *cut (!), assert, asserta, assertz* nor those predicates explicitly defined for I/O, like *read*, *write*, *print*. Among predicates for list processing, you can use *append*, *length*, *member*. All other predicates you need for list processing must be defined by yourself.

Submission: one zipped file called YourLastName_YourFirstName containing

- The source code file.
- A readme file in plain text or pdf format, providing the following information:
    - The meaning of each symbol you introduced. Both the function symbols and the predicate symbols should be explained together with the arguments.
    - A list of all the function symbols that you introduced (if any).
    - A list of all the predicate symbols that you introduced.
    - The query to check whether a binary tree is symmetric.