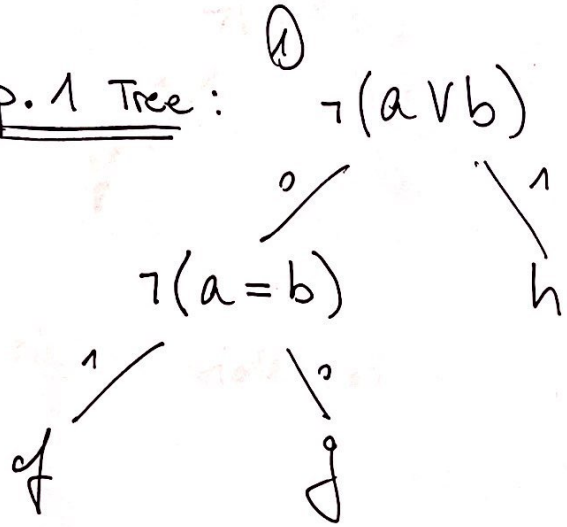
Appendix 1

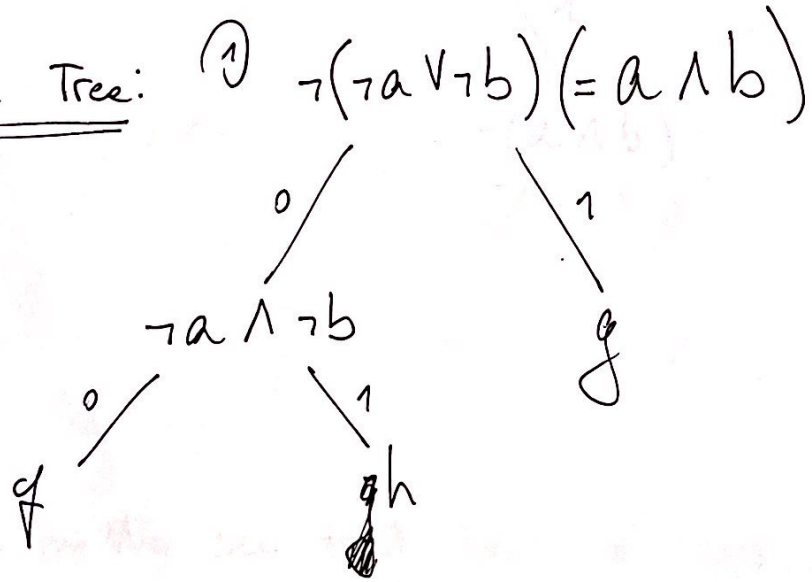
Expression 1 : $\neg(a \vee b) \,?\, h : \neg(a = b) \,?\, f : g$

Expression 2 : $\neg(\neg a \vee \neg b) \,?\, g : (\neg a \wedge \neg b) \,?\, h : f$

Exp. 1 Tree:

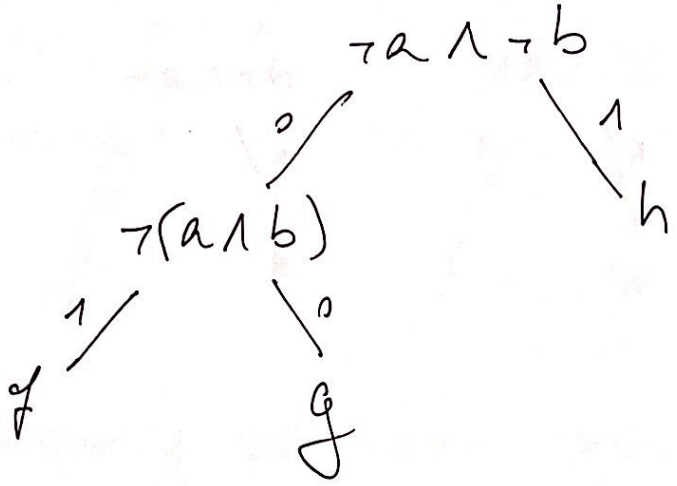


Exp. 2 Tree:

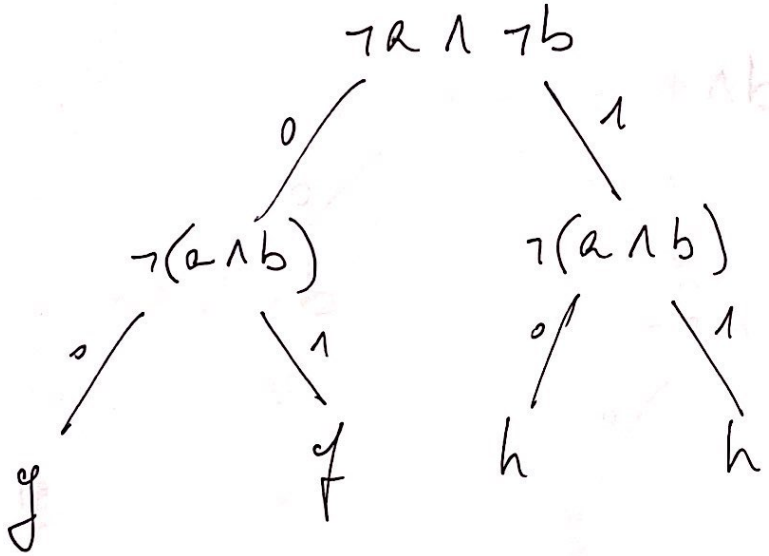


Let's first expand some of the logical trees on Exp. 1 Tree so that we can try and see if there is a way to reconstruct Exp. 1. Tree into that of Exp. 2 Tree.

② $\neg(a=b)$ can be translated to $\neg(a \wedge b)$, and $\neg(a \vee b)$ is broken into $\neg a \wedge \neg b$:

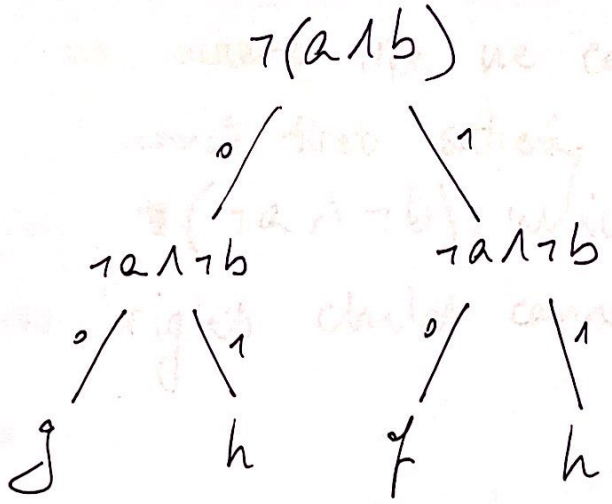


③ right side of the tree can be further expanded:

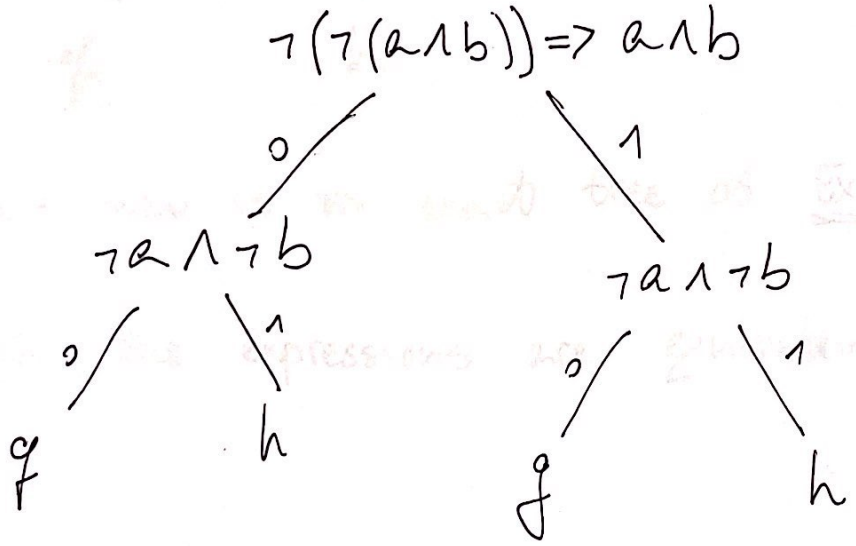


④ ~~since we~~ We see that both childrens are the same. What is more, since h can only be reached when $\neg a \wedge \neg b$ and $g \wedge f$ when not $\neg a \wedge \neg b$, we can switch the order of the nodes:

④



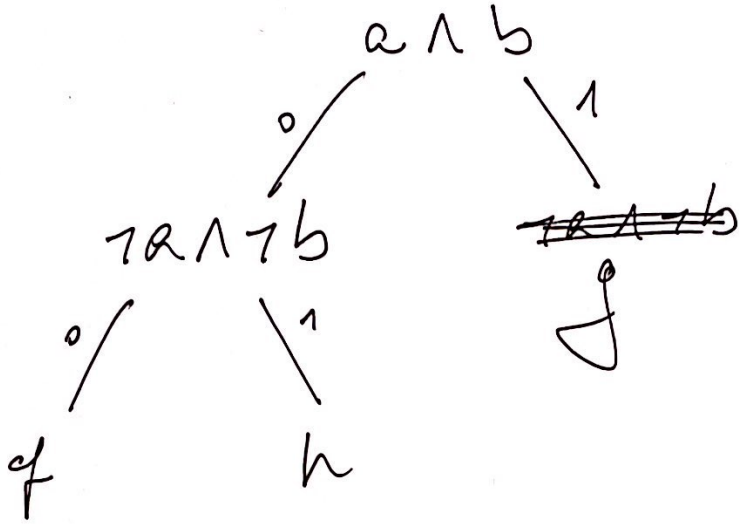
⑤ Comparing our current tree with the one of Exp. 2, we see that they are getting more similar. Let us now negate the root node.



*here we had to swap the child nodes as a consequence of negating the parent node.

⑥ From our current tree we can observe that we cannot first satisfy $a \wedge b$ and then satisfy $\Diamond(\neg a \wedge \neg b)$, which means h of the right child cannot be reached.

Hence:



This now is the exact tree of Exp. 2.

Hence, the expressions are equivalent. ■