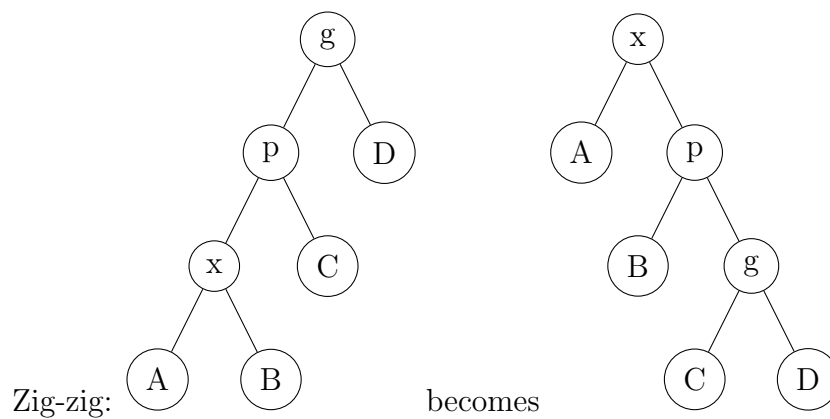
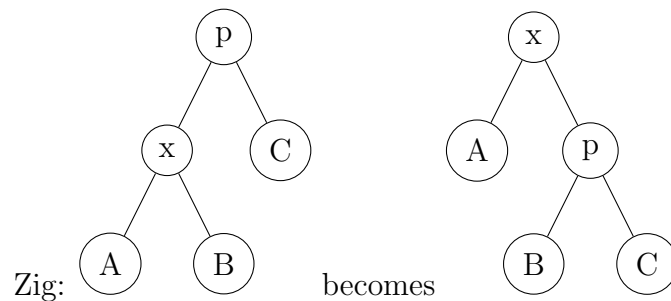
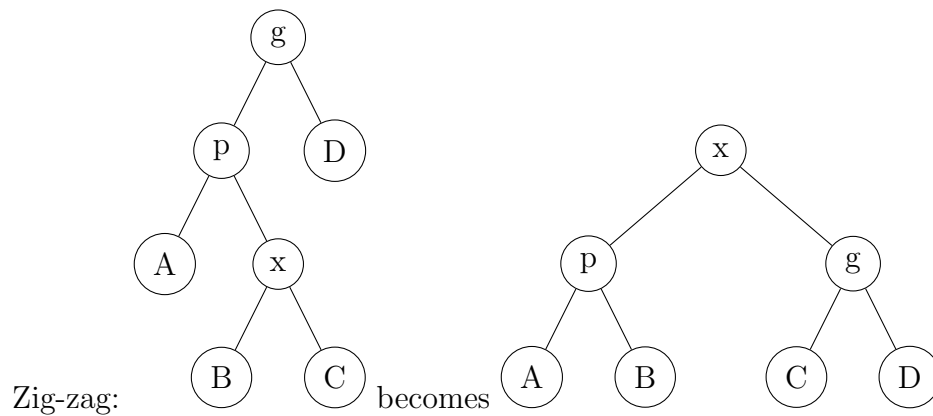


University of Waterloo
CS 466 — Advanced Algorithm
Spring 2013
Problem Set 4
Siwei Yang - 20258568

- [6 marks: Splay Trees] One feature of splaying versus the simple move to root heuristic is that in performing a splay operation (i.e. performing the appropriate splay operations to move a node to the root), no node has its depth increased by more than 2. Prove this.

Let's observe what happens when a splay operation fires off where x is the splay node, p and g are x 's parent and grandparent.





There are three other cases symmetrical to what listed above, but we will omit them for the sake of simplicity.

Through observation, we find for each splay step,

- all nodes affected becomes x 's descendents
- the nodes started as x 's descendents will not increase their depth
- the nodes becomes x 's descendents will increase their depth at most by 2

And we know any node can become x 's descendents at most once, and their depth won't increase after that. Thus, **during a full splay operation, no node has its depth increase by more than 2.**

2. [6 marks: Close to 2-SAT] Suppose you have a Boolean formula in CNF (conjunctive normal form) with n variables and also n clauses. However $\lg n$ clauses have three literals (variables or their negations) and the remaining ones have just two. Sketch an efficient algorithm to determine whether the formula is satisfiable.