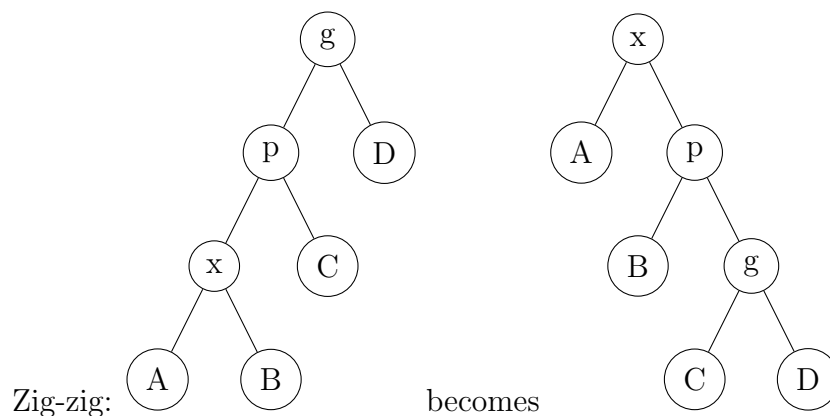
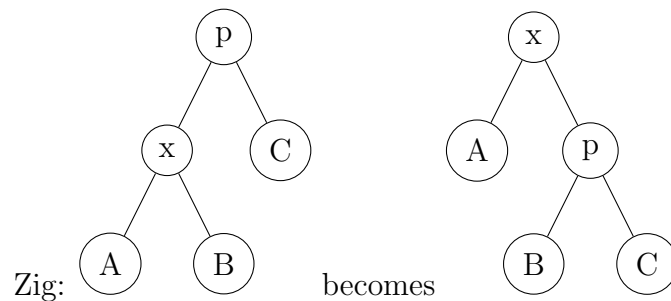
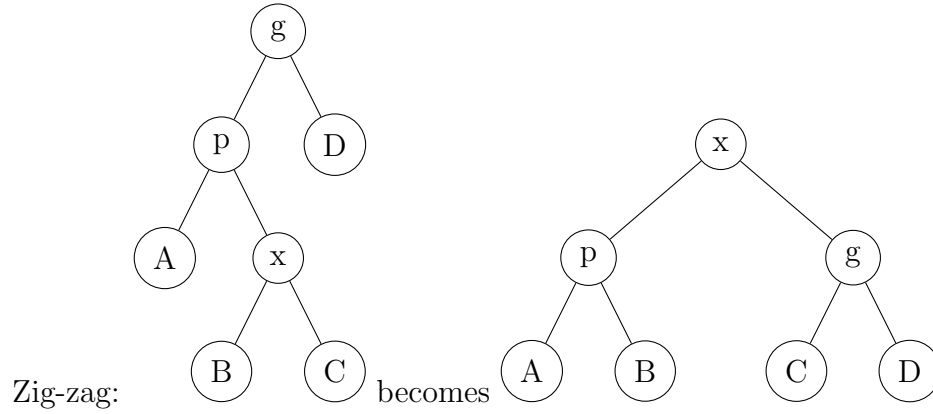


University of Waterloo
CS 466 — Advanced Algorithm
Spring 2013
Problem Set 4
Siwei Yang - 20258568

1. [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.

The algorithm to solve the formula is to divide the problem into 3-SAT and 2-SAT problems. Then solve the 2 problems individually.

Let the original problem be F . The sub-problem only has 3 literals clauses is F_{3-SAT} . The sub-problem only has 2 literals clauses is F_{2-SAT} .

$$F = F_{3-SAT} \wedge F_{2-SAT} \quad (1)$$

First tackle F_{3-SAT} by enumerating all possible variable assignments of the variables in the formula. Consider the number of variables need to be covered:

$$|v(F_{3-SAT})| \leq 3 * \lg n \quad (2)$$

Thus, the size of the enumeration:

$$2^{|v(F_{3-SAT})|} \leq n^3 \quad (3)$$

For each enumeration, it's a linear time operation to check if the assignment is valid. In case of a valid assignment, substitute the assignment into F_{2-SAT} to arrive at F'_{2-SAT} . Use the linear algorithm taught in class to determine if F'_{2-SAT} is satisfiable. If F'_{2-SAT} is satisfiable, then F is satisfiable. Otherwise, move on to the next assignment for F_{3-SAT} . Unless at least one assignment creates a satisfiable F'_{2-SAT} , the original problem F is not satisfiable. **The algorithm is a composite of three linear time operation, thus the algorithm is still linear time.**