

## Exercise 1 – Advantage of backward over forward chaining

Backward chaining is goal-driven and uses DFS and starts from the conclusion and traverses through each inference rules until the conclusion can be made true. Thus, one may avoid having to search parts of the tree and therefore reduce running time. On the other hand, forward chaining is data-driven and uses BFS for all true facts and traverses through each inference rule, where the given fact is its literal. This would, therefore, require one to exhaustively traverse through the tree until a conclusion has been reached.

Based on that, if one has a clear conclusion, then it may be better to use the backward chaining as it is goal-driven and may there be faster for such cases. Whereas if you have a lot of data but do not necessarily the final goal or conclusion, then it would be better to use forward chaining and derive conclusions that might not be known yet

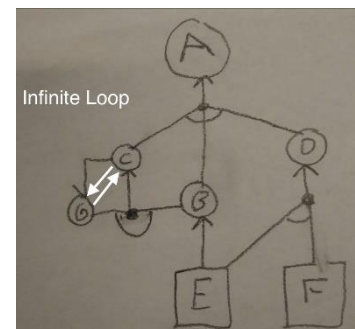
## Exercise 2 – Professor Smart's Backward chaining on KB entails A

The algorithm will go into an infinite loop and thus A will never be true. The reason for this because A depends on C and C depends on G. Since C and G depends on each other, the algorithm will not be able to return since both are an inference rule and not a fact. Basically G calls **CHECK-ALL** on C and C calls **PL-BC-ENTAILS** G. This will continue forever.

Therefore it can be concluded that KB does not entail A since C and G will never be true.

To fix this problem, one must keep track on all visited clauses and their states such that one may return false if a clause is being revisited and nothing has changed.

```
Function boolean Check-All(KB,premise){  
    Foreach p in premise{  
        if(!visited(p)){ continue }  
        return PL-BC-Entails(KB, p)  
    }  
}
```



## Exercise 3 – Is Smart's algorithm efficient

No because the worse case running time of Smart's algorithm is  $O(N^2)$  for N clauses, the function in PL-PC-Entails function loops over N clauses. And for each N clause that a looped over, the CHECK-ALL function will loop over N-Clauses.

One may use dynamic programming and cache all evaluated clauses. This may reduce running time because one avoids reevaluating these clauses again but refer to the cache instead.

