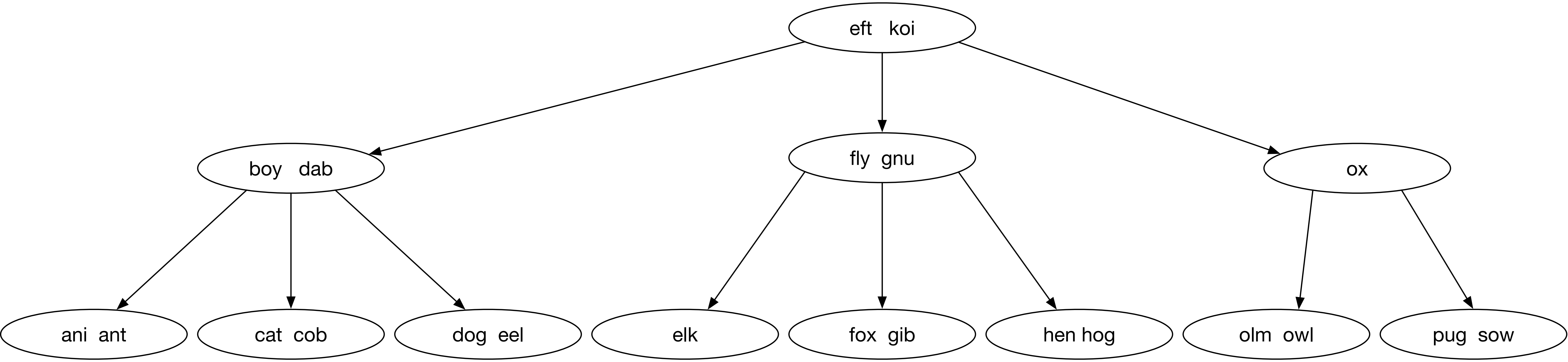
Activity: B-tree Operations

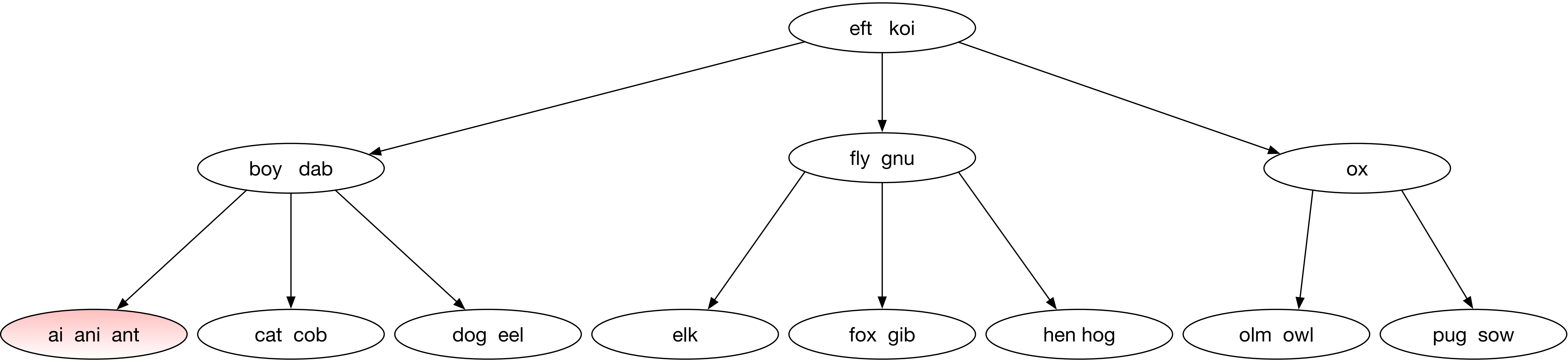
How do we insert 'ai'?

- Rules:
  - Prefer left rotate over right rotate over split / merge



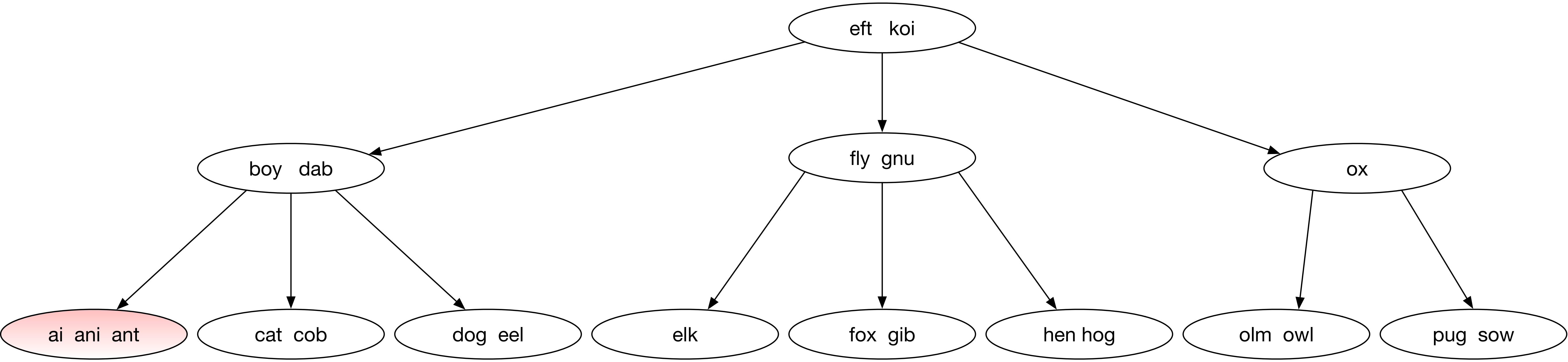
Solution

- Insert into leaf and diagnose an overflow



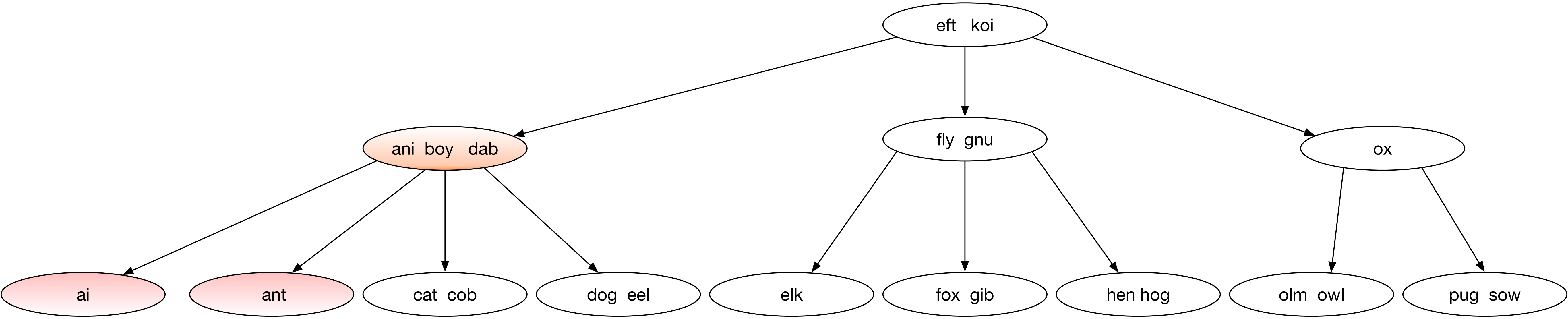
Solution

- Only neighbor is at capacity, need to split



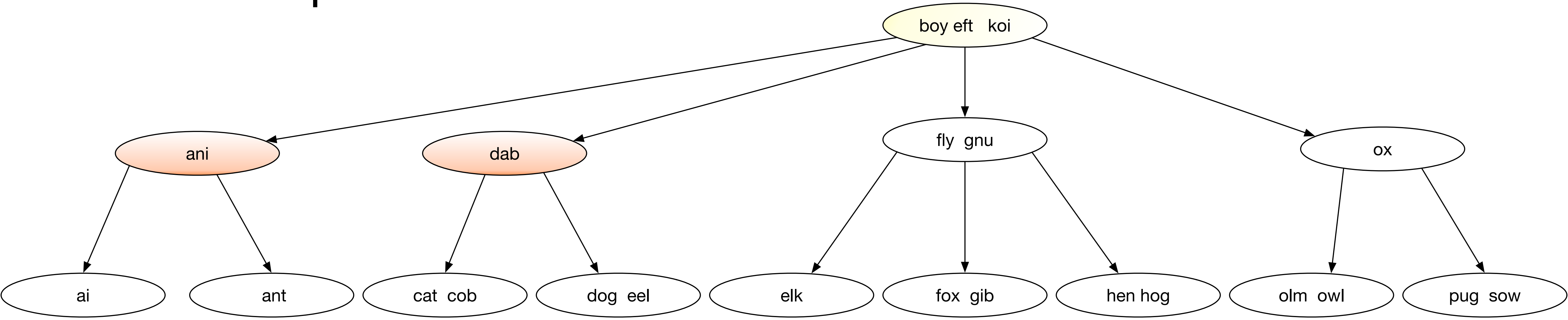
Solution

- The split results in another overflow



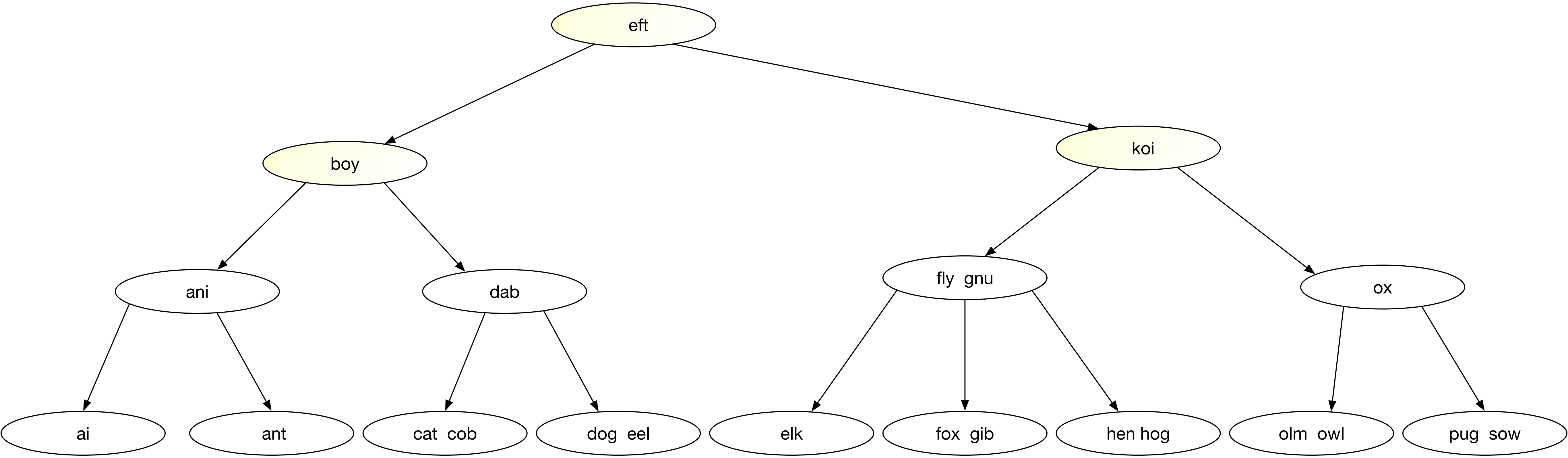
Solution

- The only neighbor is at capacity, therefore split
  - The parent is at capacity, has no neighbor, so we need to split.



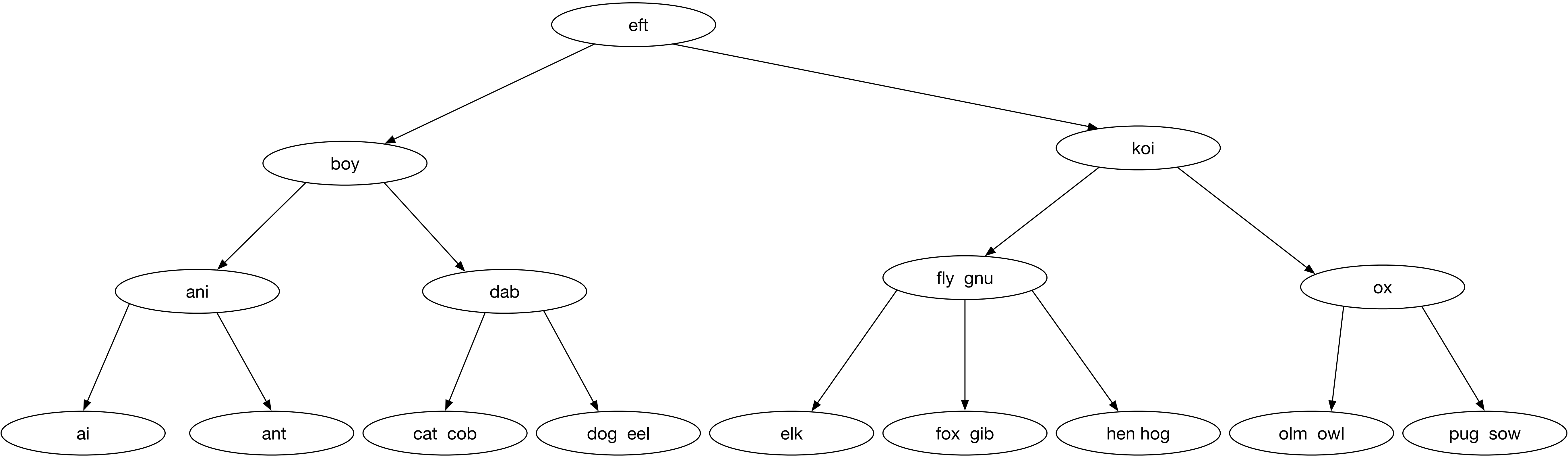
Solution

- Final result



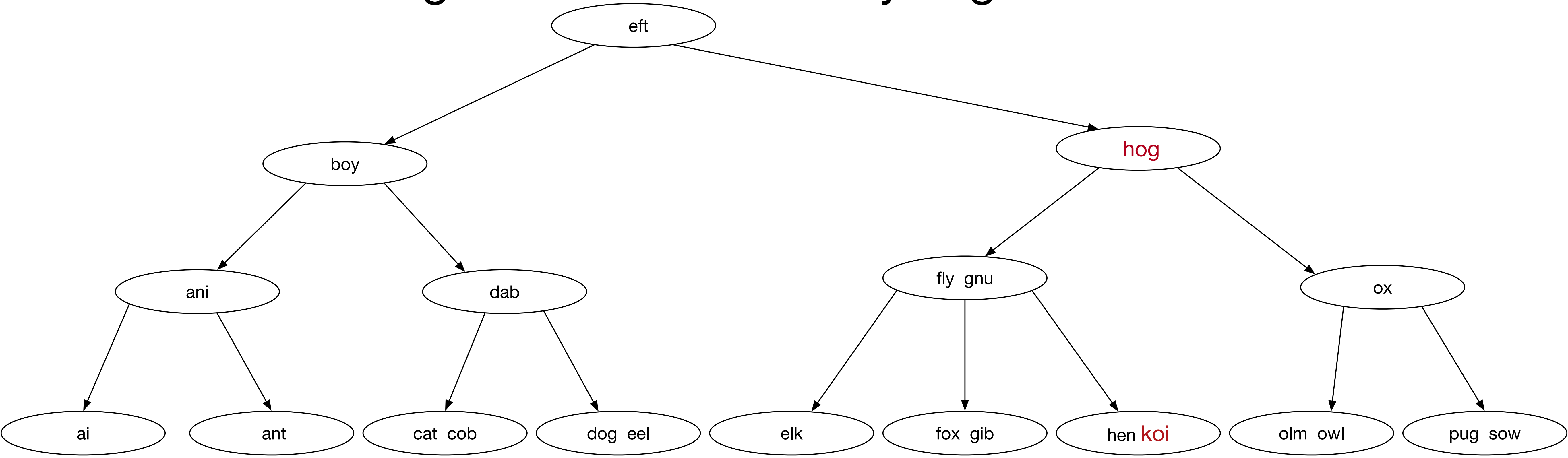
Problem

- Delete 'koi' from this tree
    - Rules: Use always the predecessor
    - Prefer left rotate over right rotate over merge



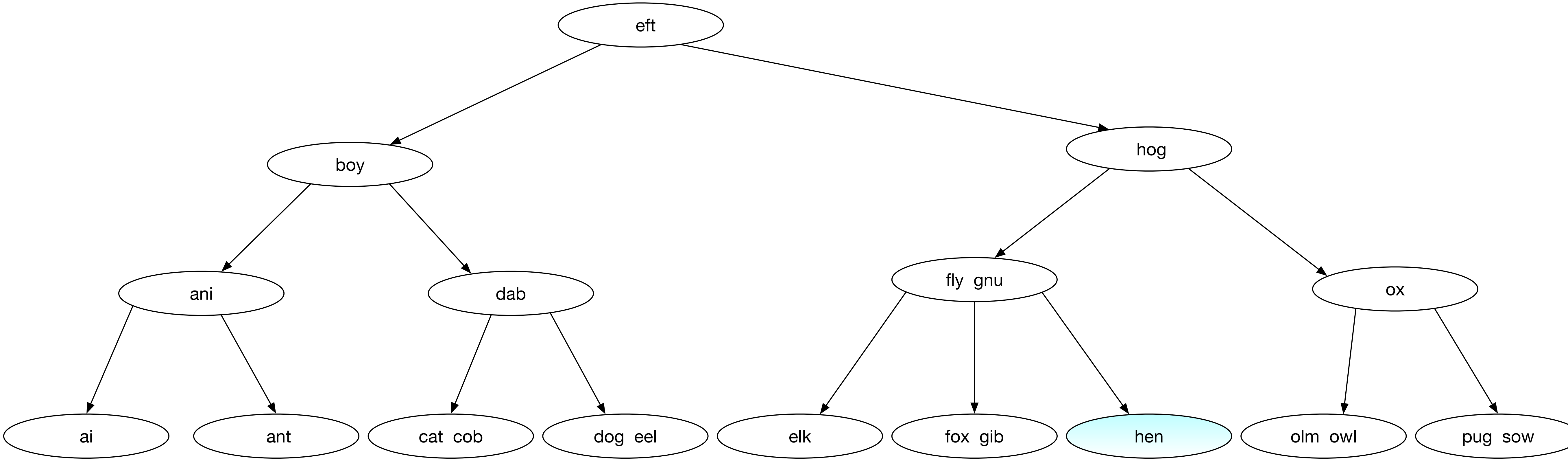
Solution

- Locate "koi", then the predecessor of "koi" and switch
- From "koi" go left and then always right



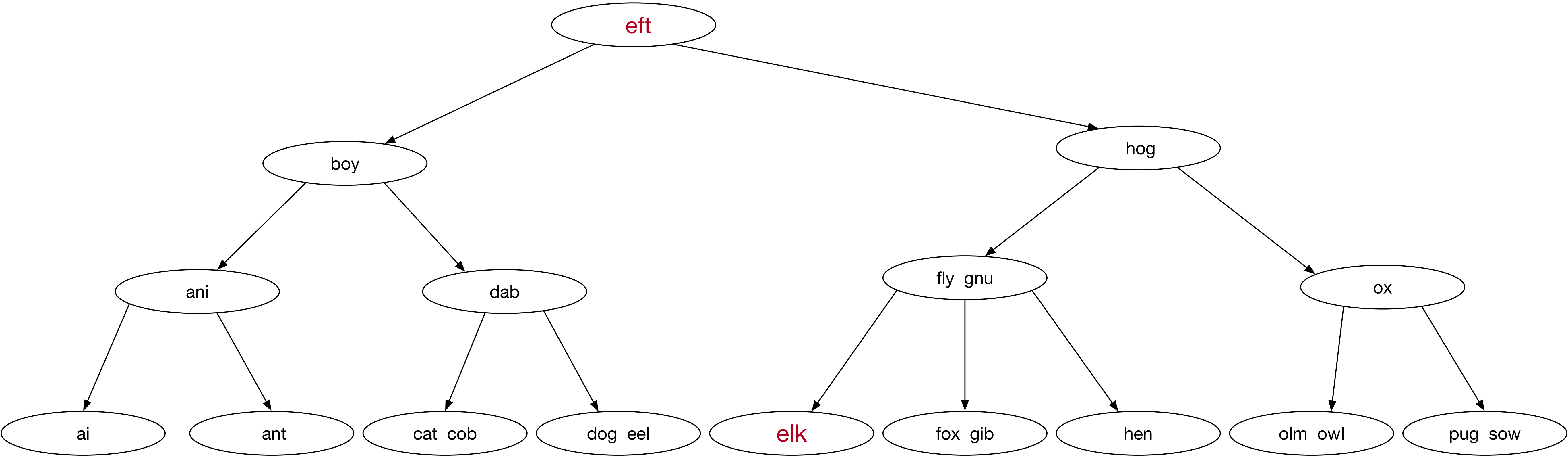
Solution

- The deletion does not lead to an underflow, so we have the final state of the 2-3 tree



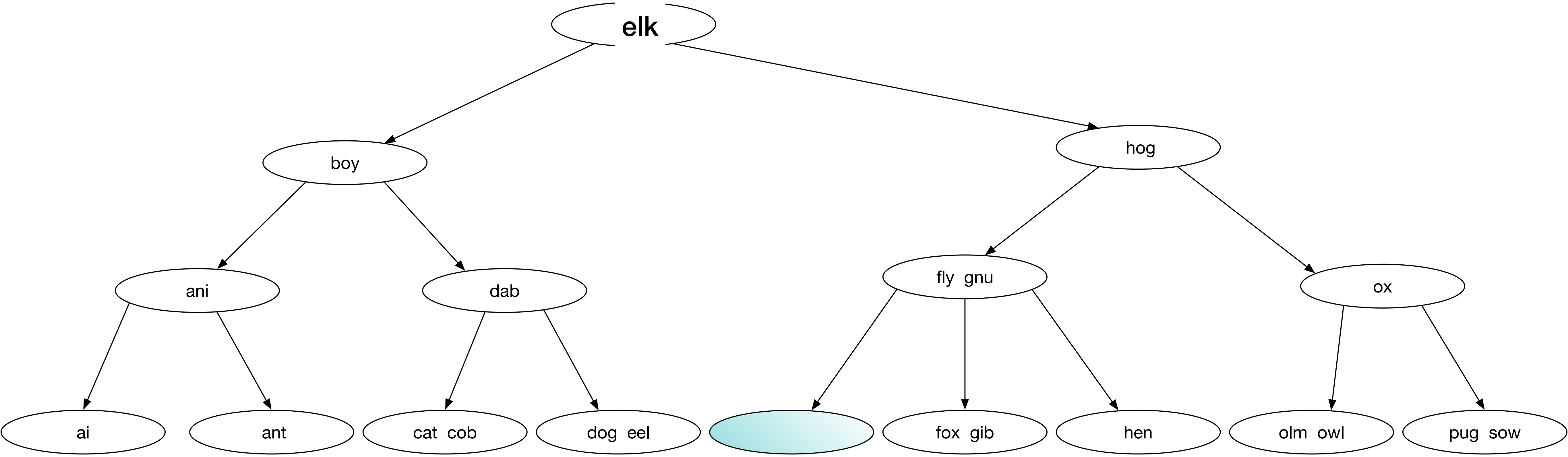
Problem

- Now use the successor to delete "eft" from the previous tree



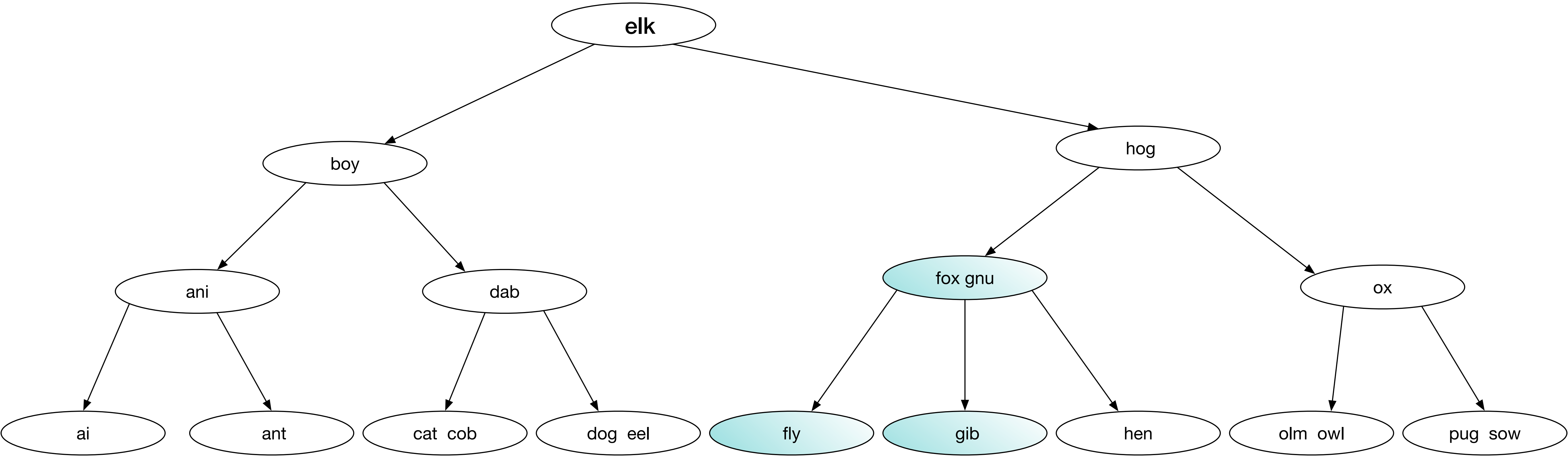
Solution

- Now the delete gives an empty (underflowing) node



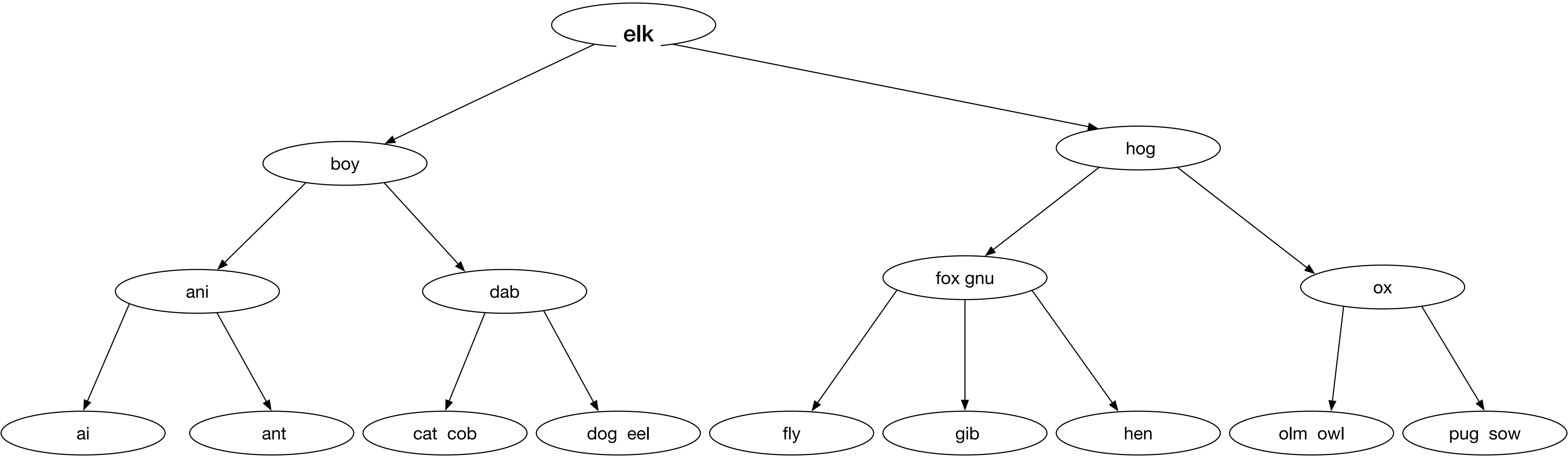
Solution

- There is only one neighbor, which is over minimum capacity, so we rotate
- "fox" goes up, "fly" goes down



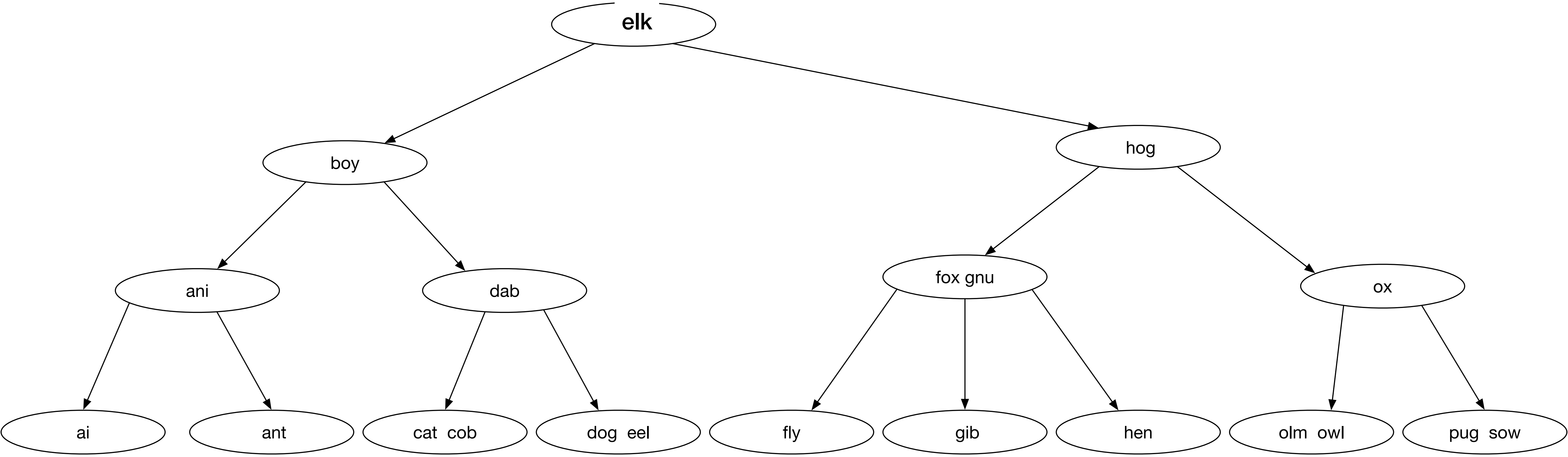
Solution

- The result is a valid 2-3 tree



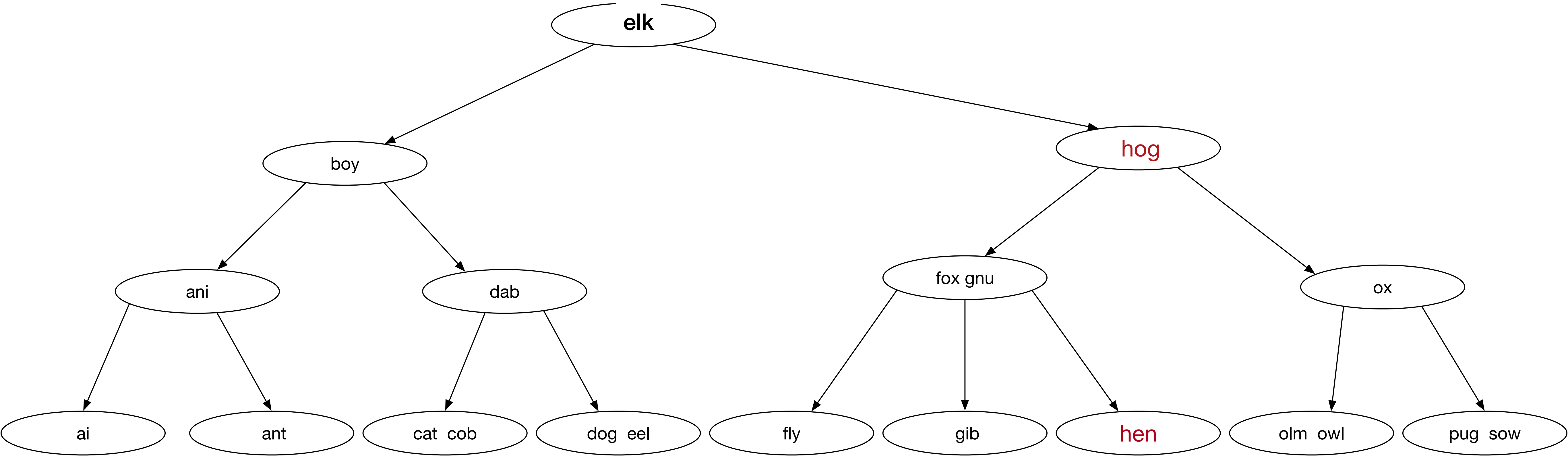
Problem

- Delete hog using the predecessor



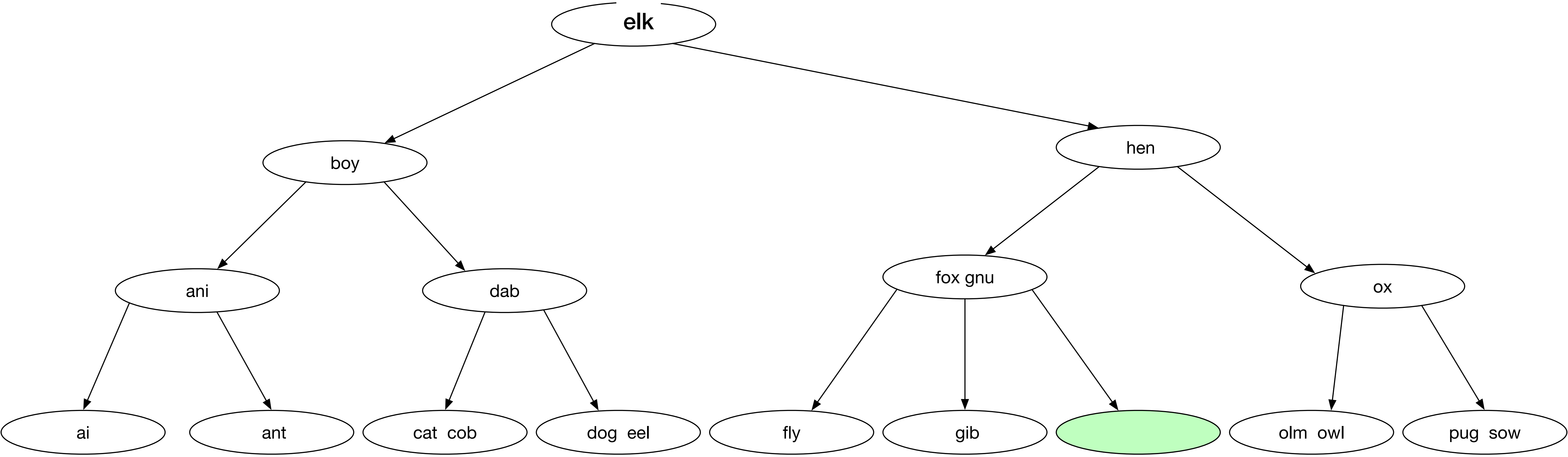
Solution

- We find 'hog', then we locate the predecessor



Solution

- We switch and delete from the leaf
- The result is an empty node

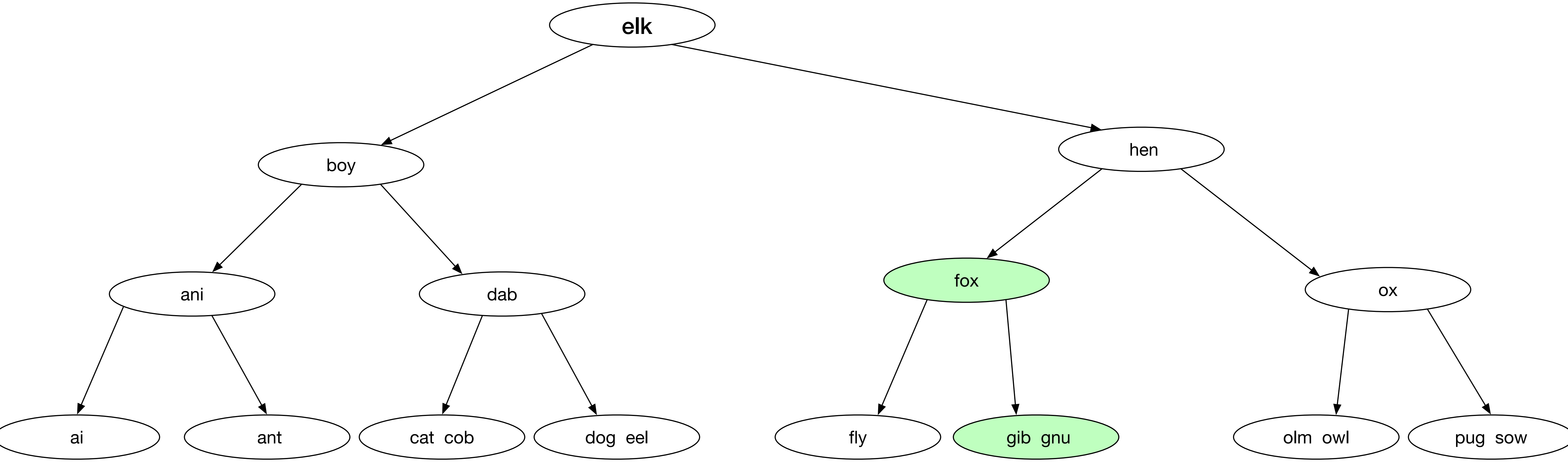


Solution

- Because we cannot do a rotate, we need to do a merge
- A merge is easiest remembered as the inverse operation of the split
    - "gnu" goes down into the united node

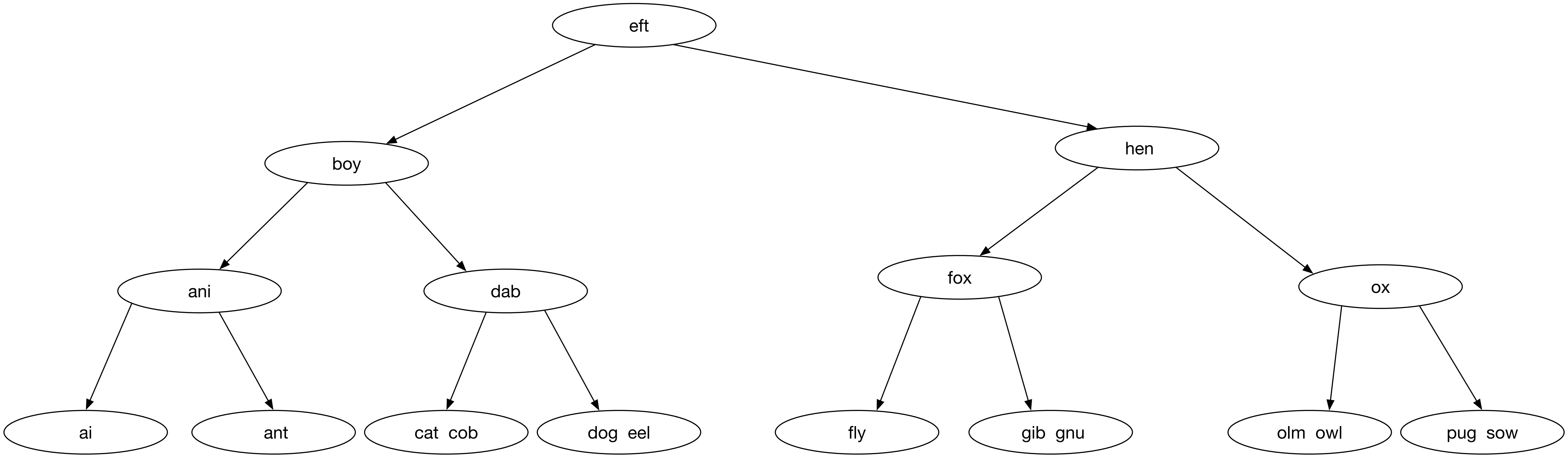
Solution

- The resulting tree complies with all requirements and we are done



Homework Problem

- Homework:
  - Delete ‘eft’, then delete ‘fox’



Rules

- Electronic submission via D2L ONLY
- Formatted
- Scanned images not accepted