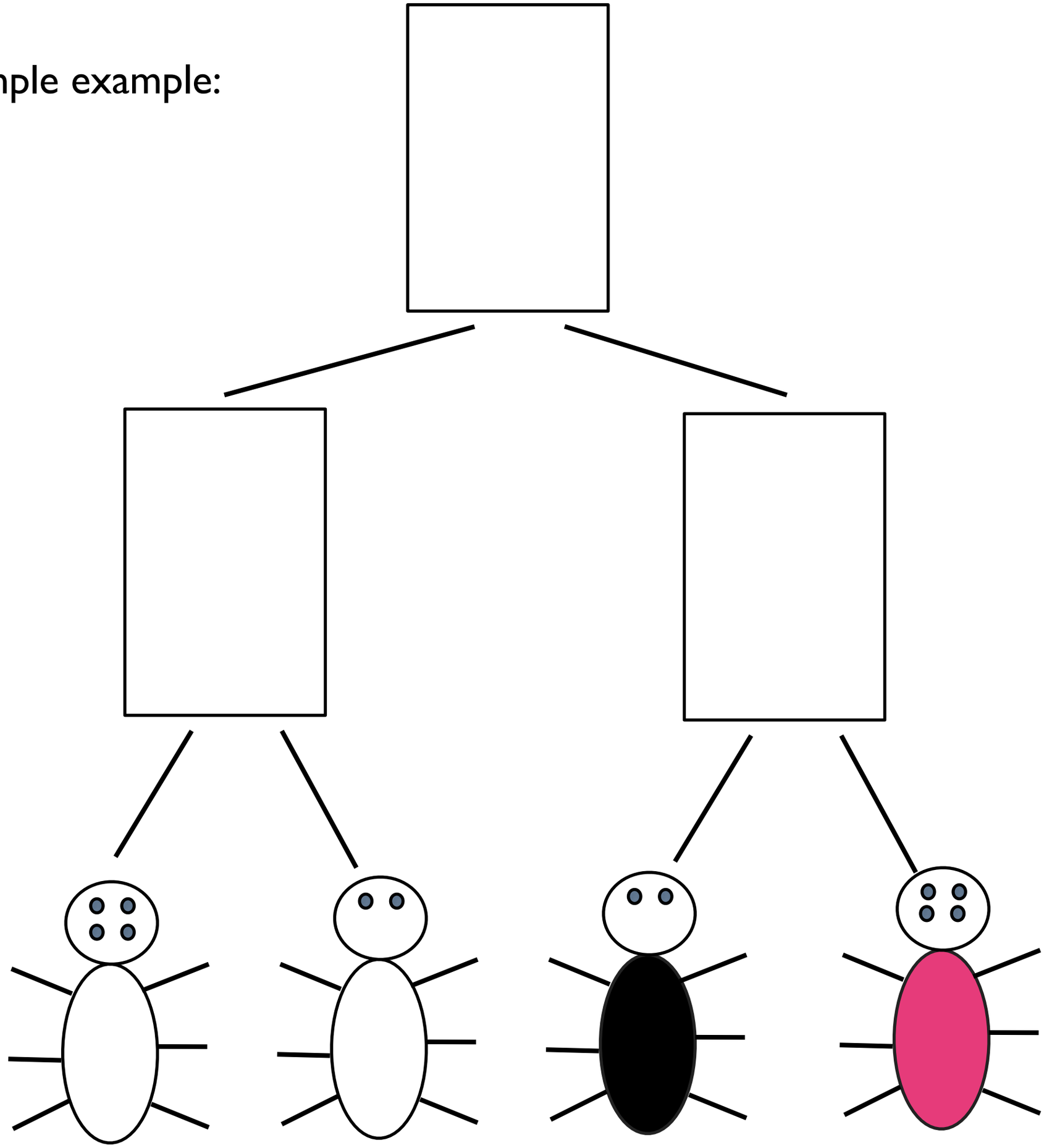
Fitch's Algorithm

CSc106 Fall 2012 Lab

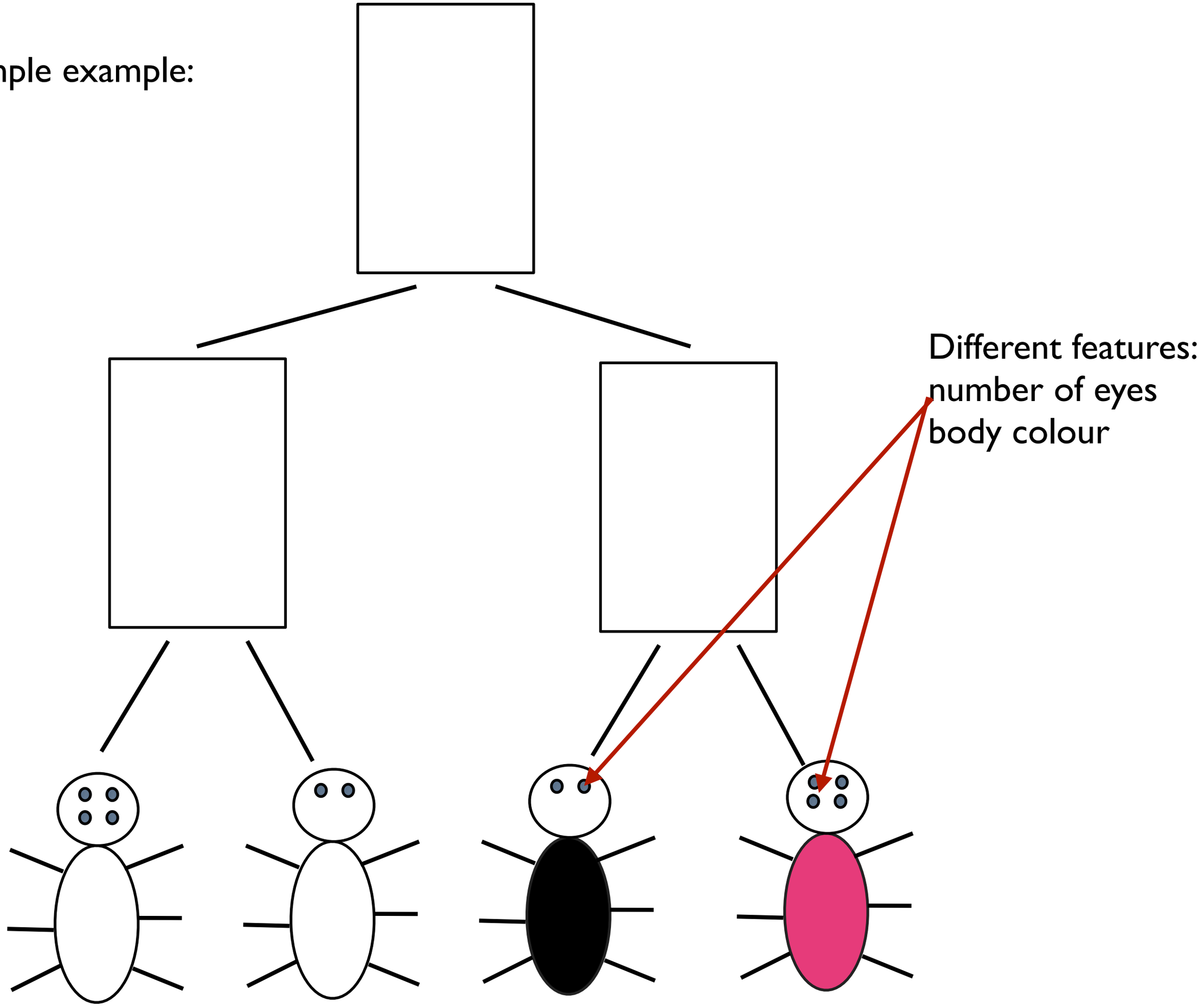
Fitch's Algorithm

- Given a set of known creatures **and**
- An ancestral tree shape
- Find a **most parsimonious** set of ancestors which could explain these creatures
- Solves the **Small Parsimony Problem**
- Can be done in **polynomial time**

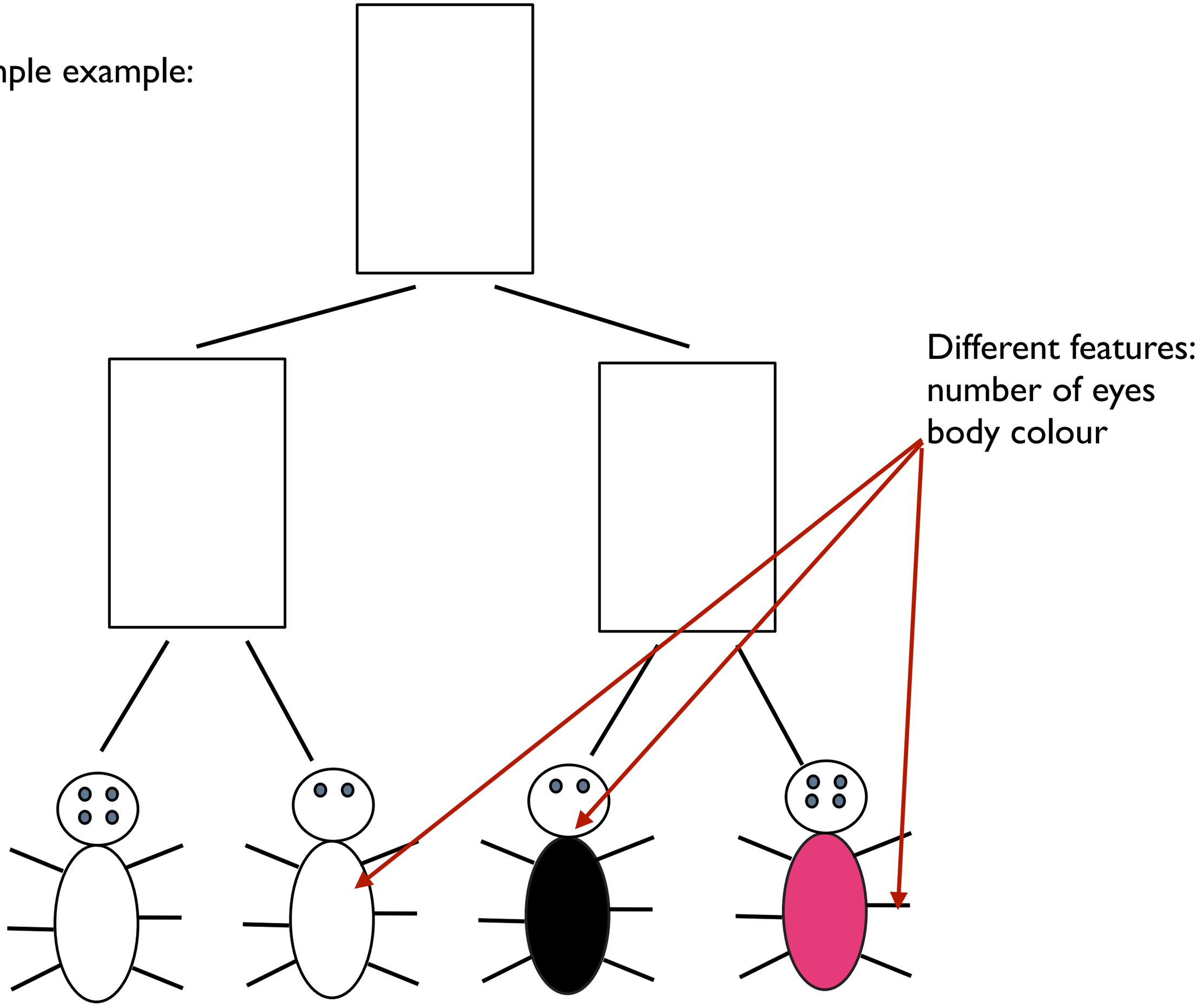
A simple example:



A simple example:



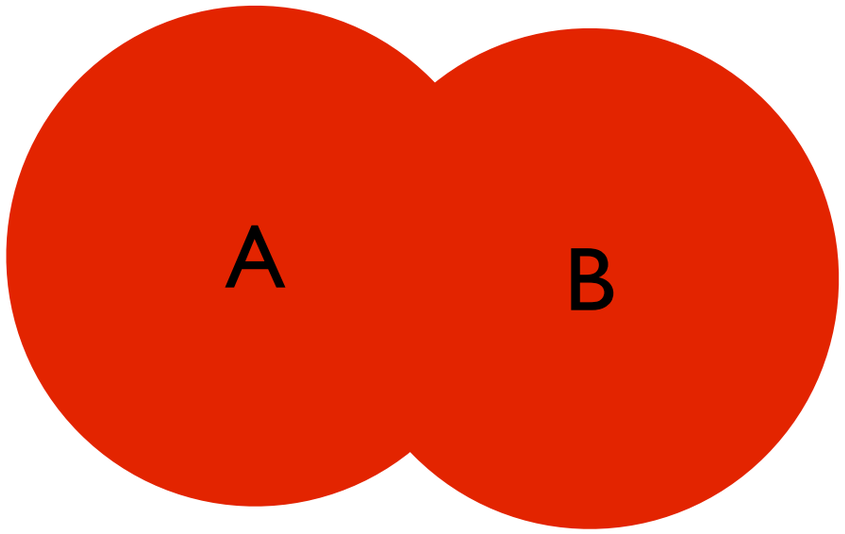
A simple example:



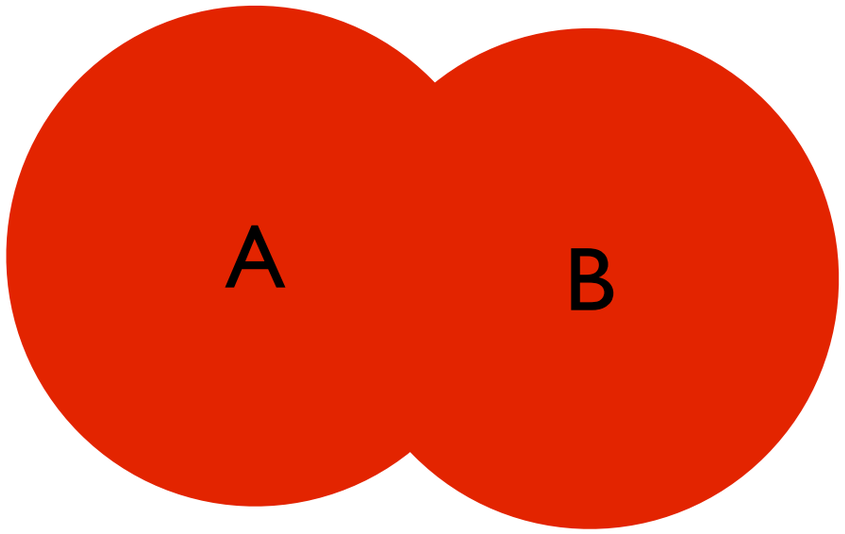
A quick review...

- Recall:
  - ***Union*** of {a,b,c} {b,c,d} is {a,b,c,d}
  - ***Intersection*** of {a,b,c} {b,c,d} is {b,c}
  - ***Postorder*** traversal: visit children before parent
  - ***Preorder*** traversal: visit parent before children
  - Keep left-first branch consistent!

Union and Intersection

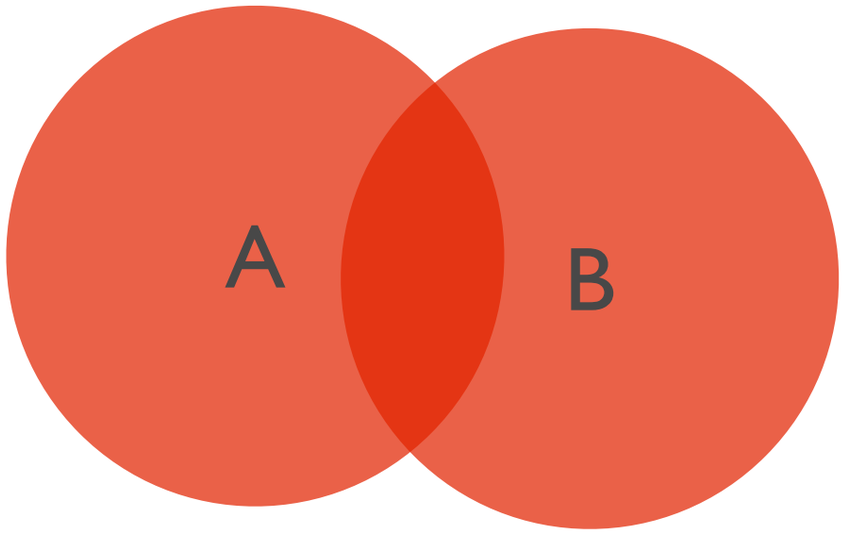


Union and Intersection



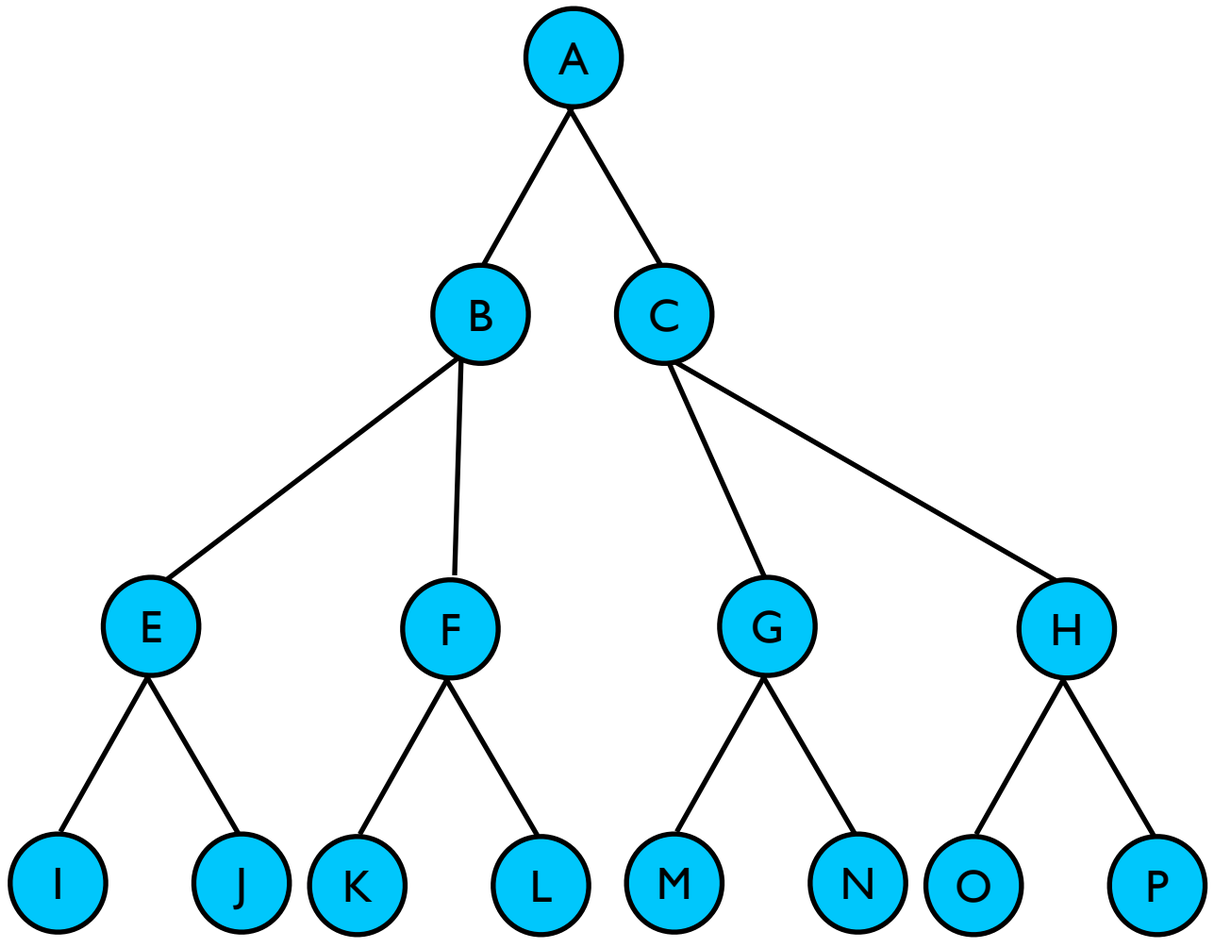
Union: All of A OR B

Union and Intersection

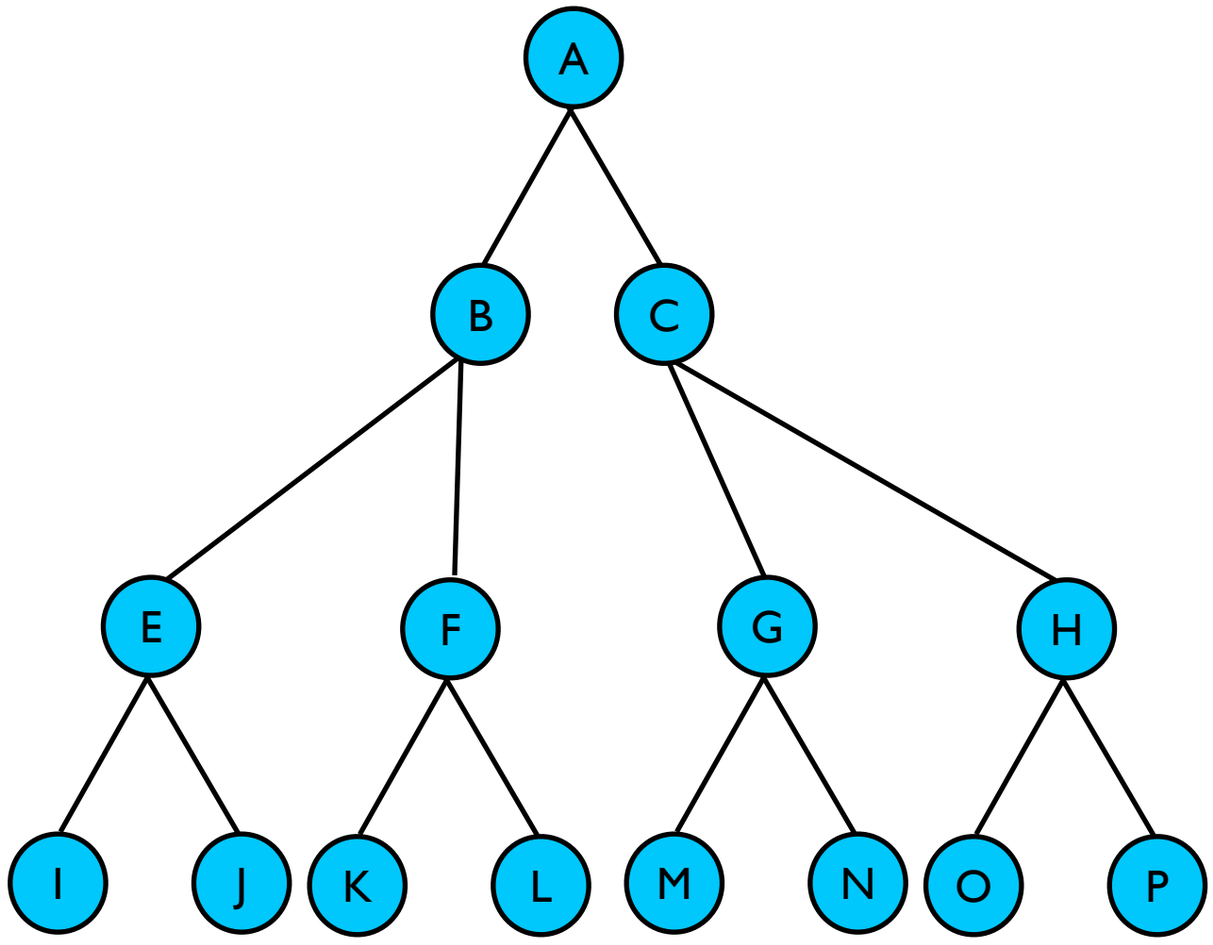


Intersection: Only in A AND B

Tree Traversals

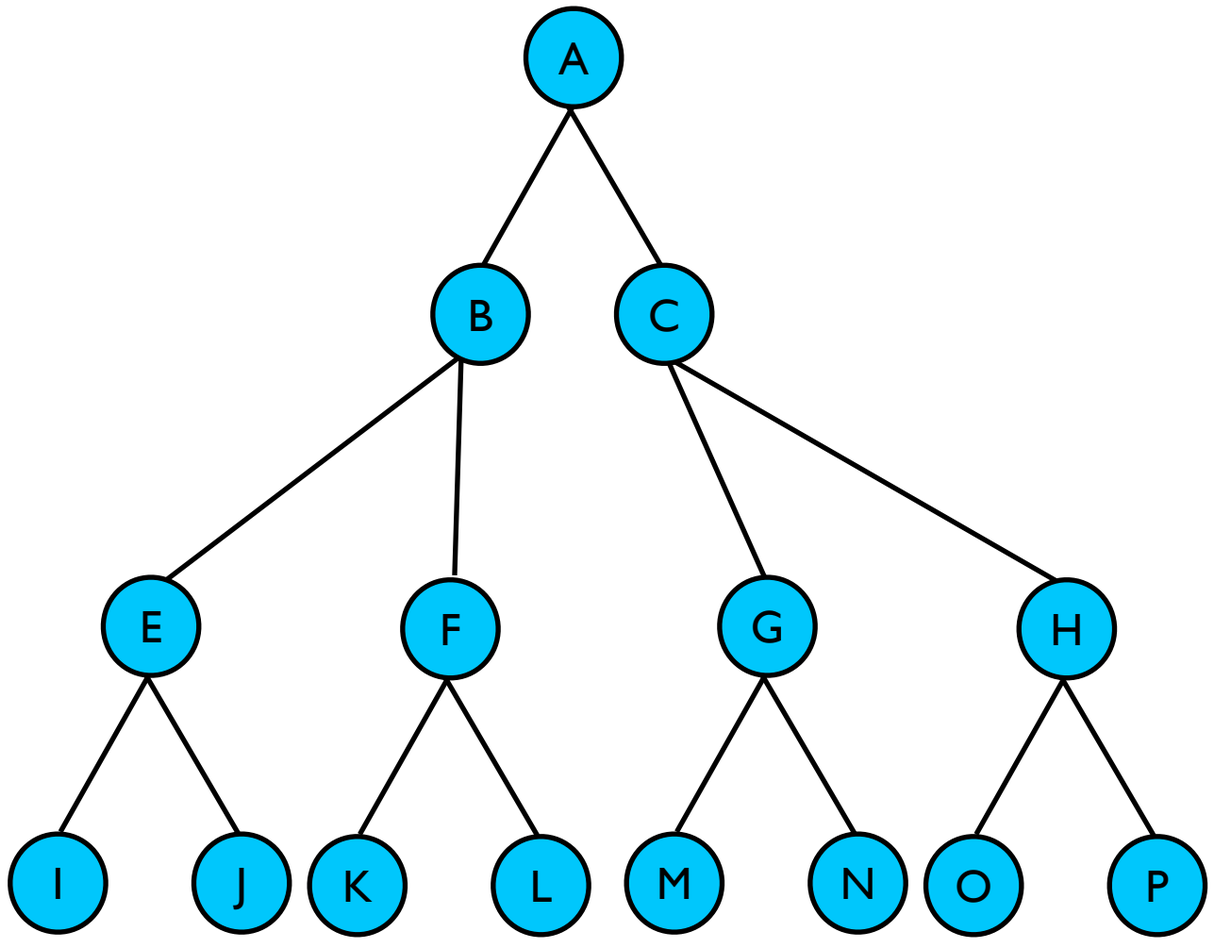


Tree Traversals



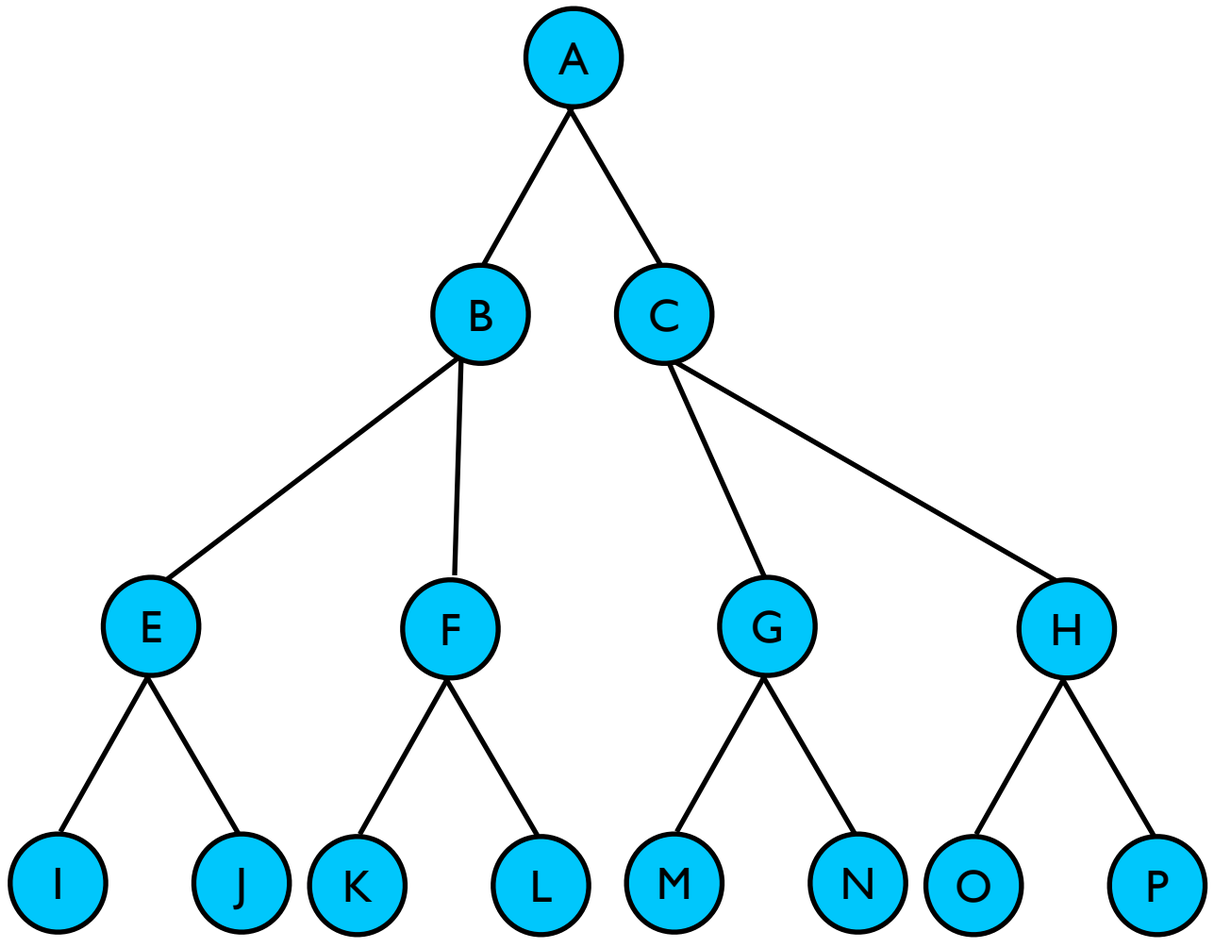
Pre-Order: Parent first, then children...

Tree Traversals



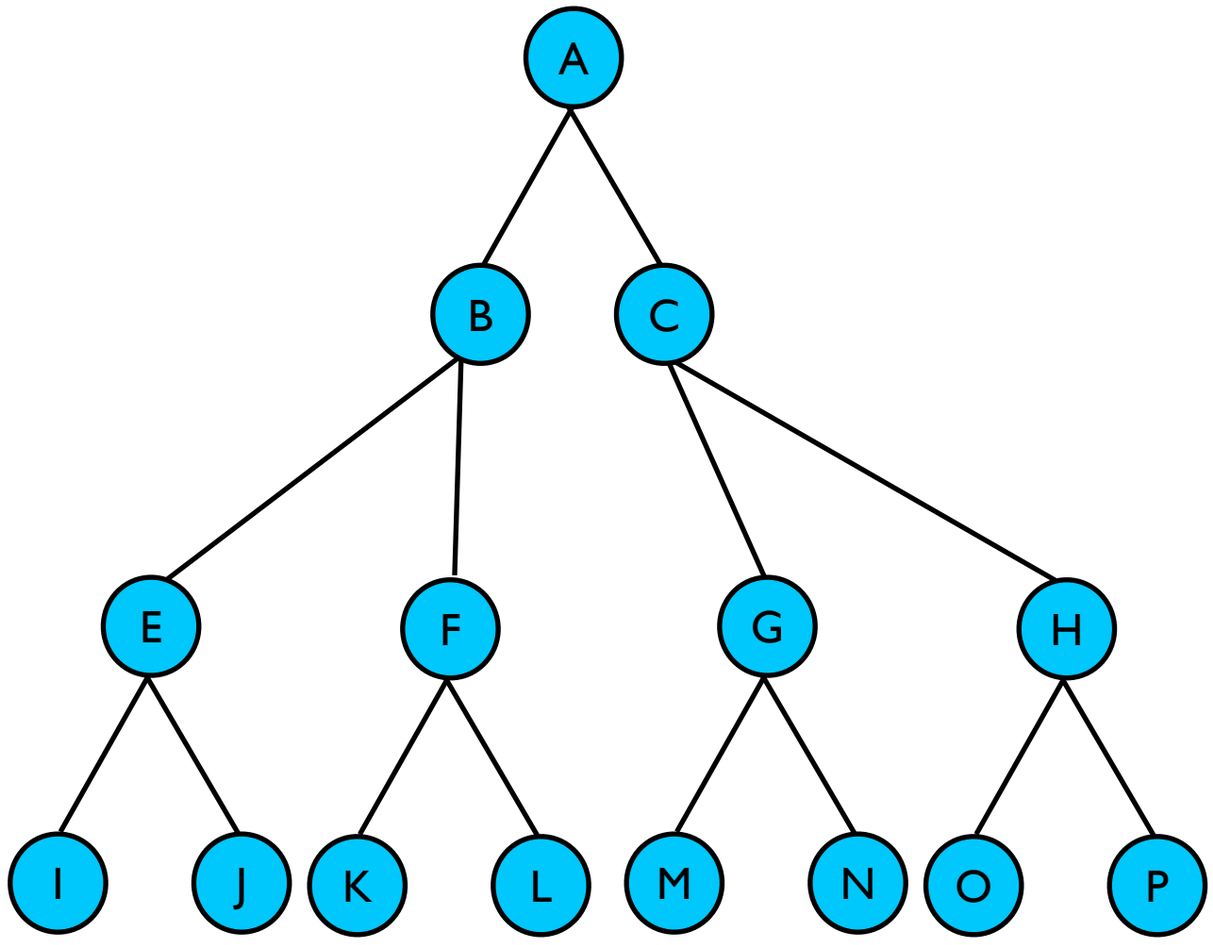
Pre-Order: A, B, E, I, J, F, K, L, C, G, M, N, H, O, P

Tree Traversals



Post-Order: Children first, then parent...

Tree Traversals



Post-Order: I, J, K, L, M, N, O, P, E, F, G, H, B, C, A

Fitch's Algorithm

LABEL TREE

Traverse tree in post-order, for each node:

if (features of the two children *intersect)*

label the parent with *intersection* of its children's features(s)

else

label with the parent with *union* of children's feature(s)

ASSIGN FEATURES

Traverse tree in pre-order, for each node:

if (the node is the root)

for each feature,

choose any label arbitrarily

else

if a feature's label matches the parent's feature

choose that feature

else

pick a feature from labels, arbitrarily

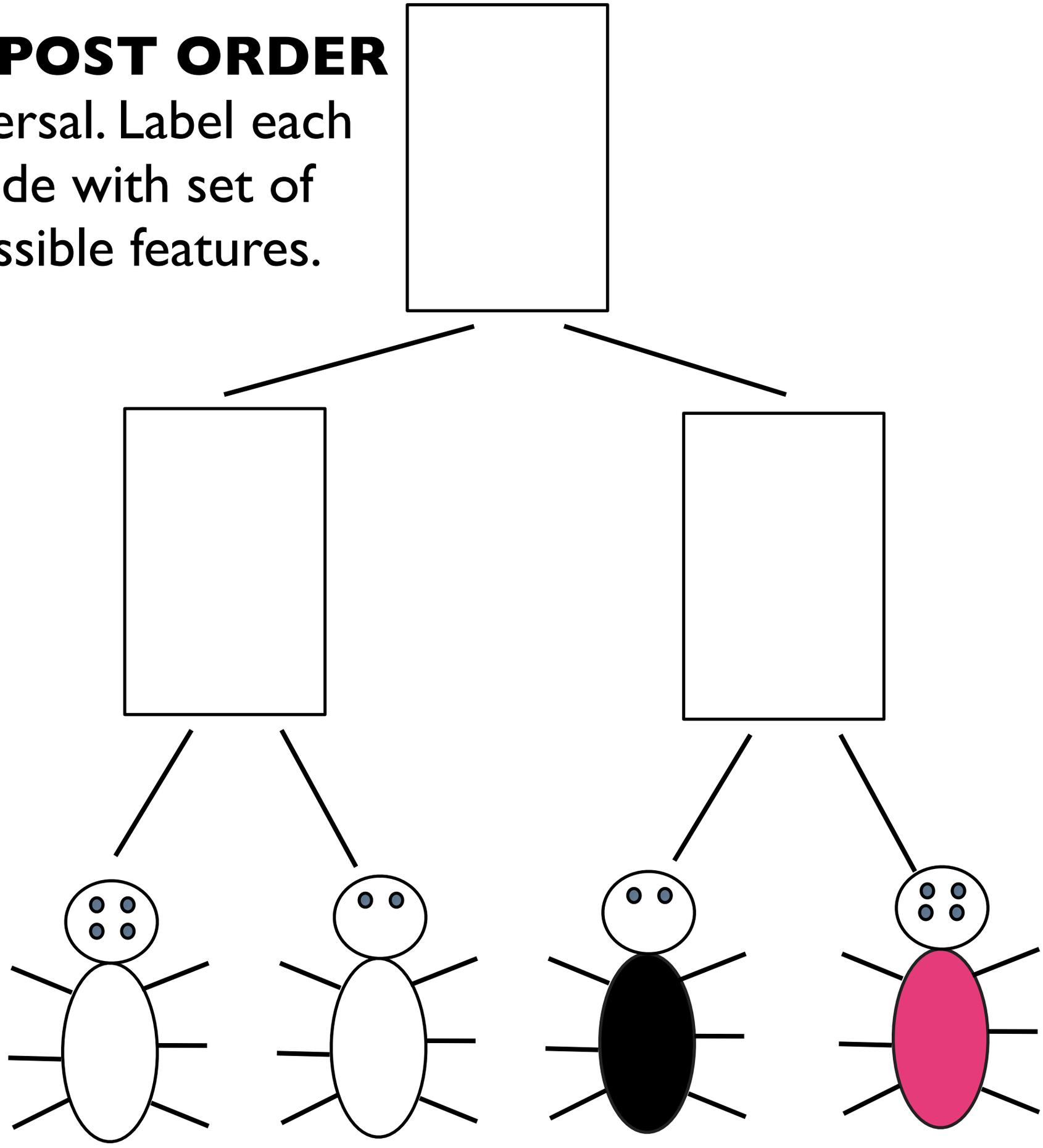
CALCULATE SCORE

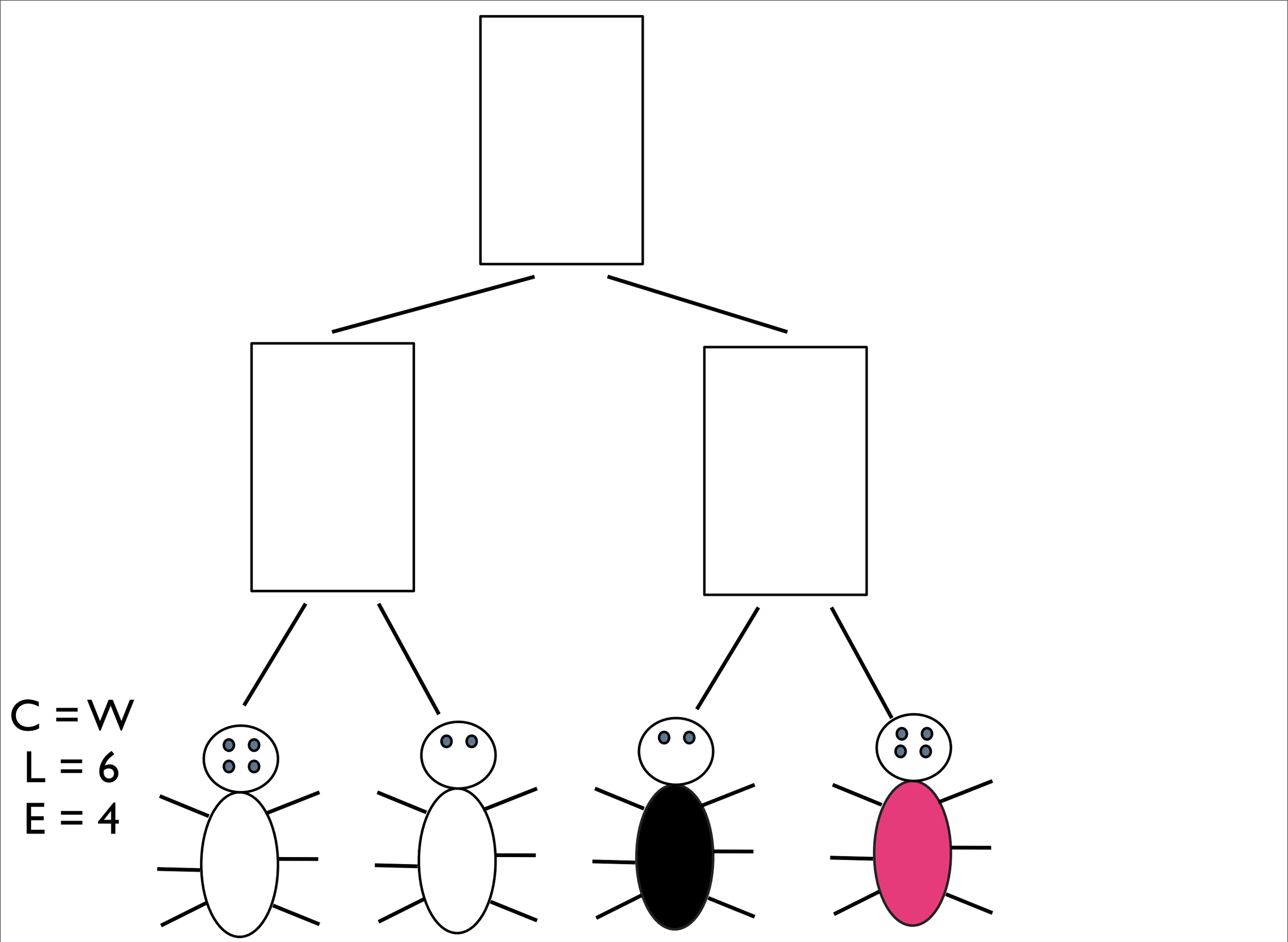
Sum the differences between all child/ancestors

Fitch's Algorithm

- Changes of different features are independent!
- Algorithm described for one feature, but can be done for all features in parallel

Do a **POST ORDER** traversal. Label each node with set of possible features.




C = W
L = 6
E = 4

C = W
L = 6
E = 4

C = W
L = 6
E = 2

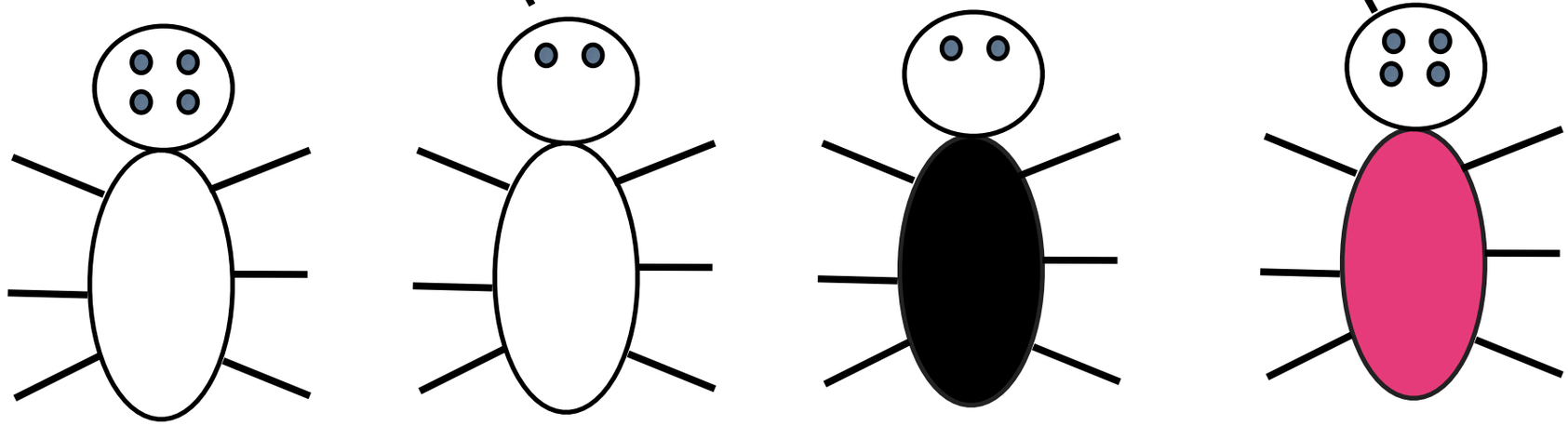
C = W
L = 6
E = {2,4}

C = W
L = 6
E = 2

C = B
L = 6
E = 2

C = P
L = 6
E = 4

C = W
L = 6
E = 4



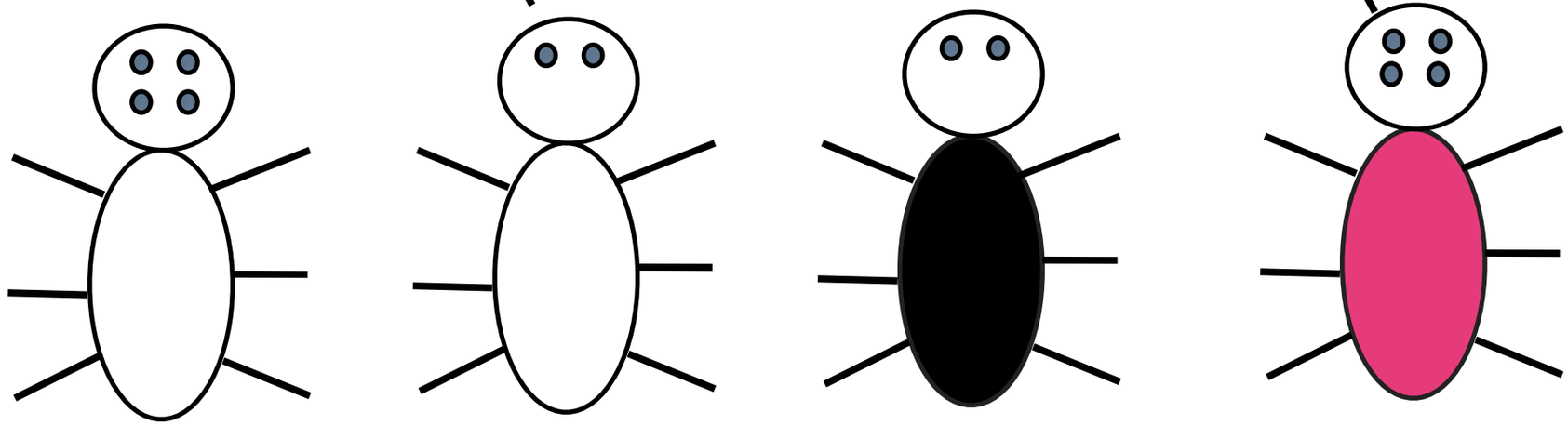
C = W
L = 6
E = {2,4}

C = W
L = 6
E = 2

C = B
L = 6
E = 2

C = P
L = 6
E = 4

C = W
L = 6
E = 4



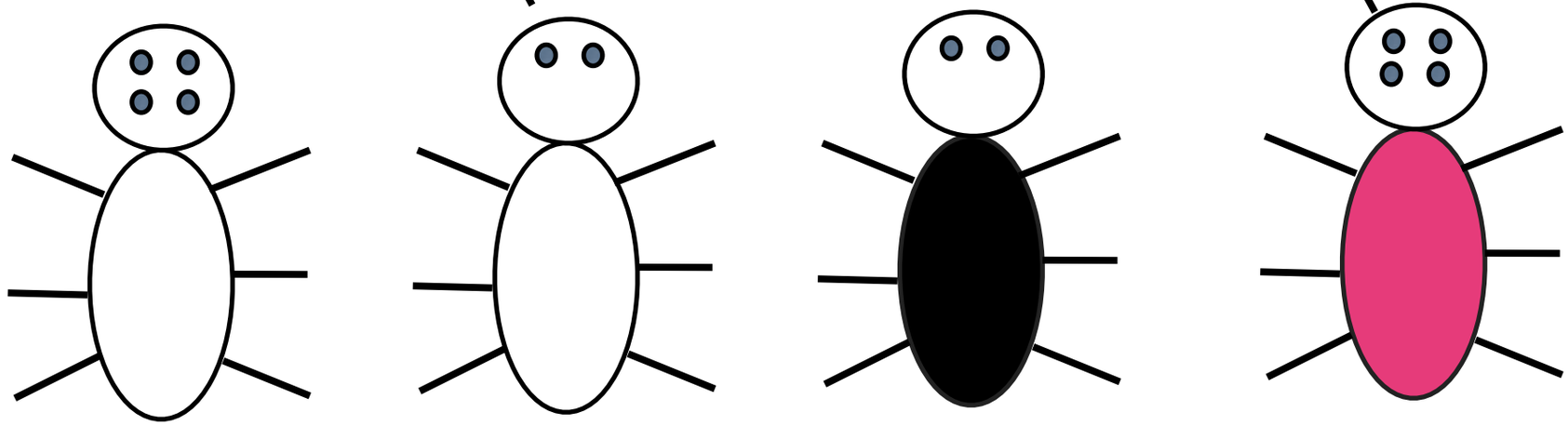
C = W
L = 6
E = {2,4}

C = W
L = 6
E = 2

C = B
L = 6
E = 2

C = P
L = 6
E = 4

C = W
L = 6
E = 4



C = W
L = 6
E = {2,4}

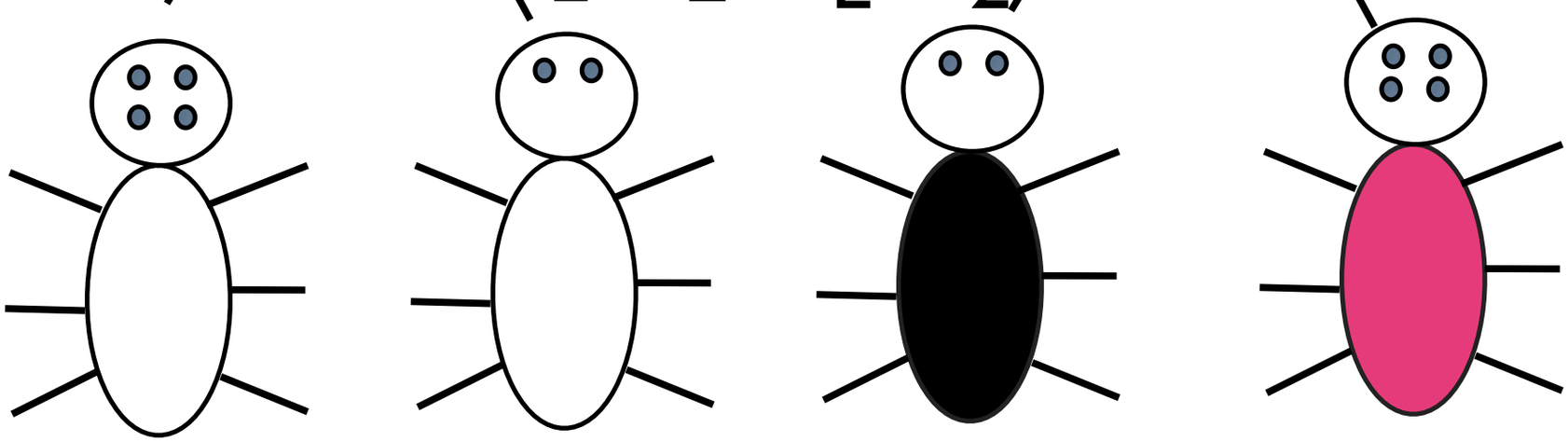
C = {B,P}
L = 6
E = {2,4}

C = W
L = 6
E = 4

C = W
L = 6
E = 2

C = B
L = 6
E = 2

C = P
L = 6
E = 4



C = {B,P,W}
L = 6
E = {2,4}

C = W
L = 6
E = {2,4}

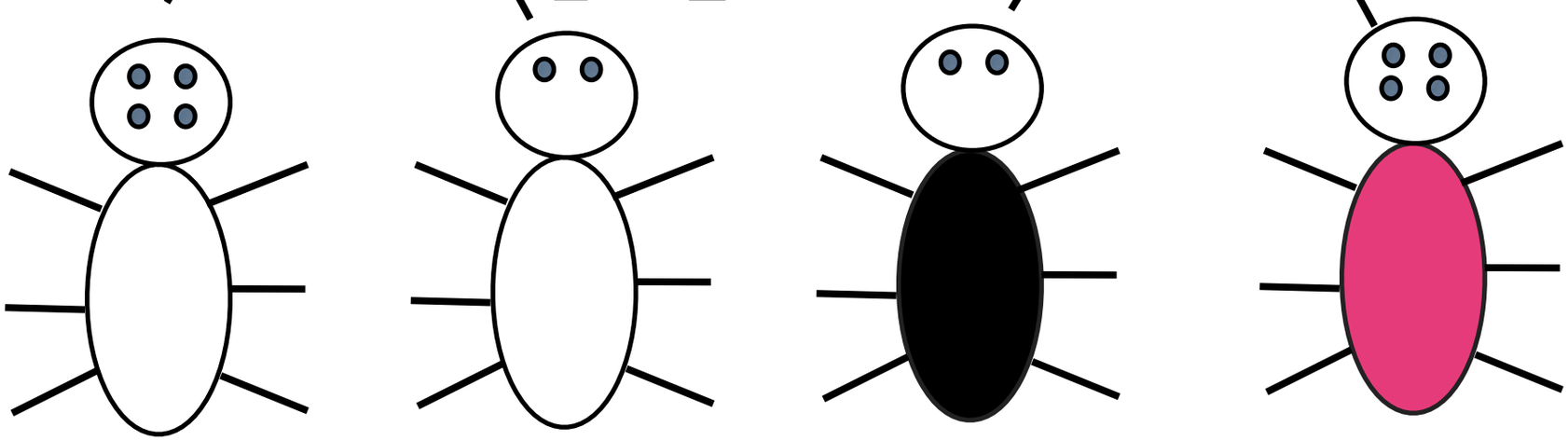
C = {B,P}
L = 6
E = {2,4}

C = W
L = 6
E = 4

C = W
L = 6
E = 2

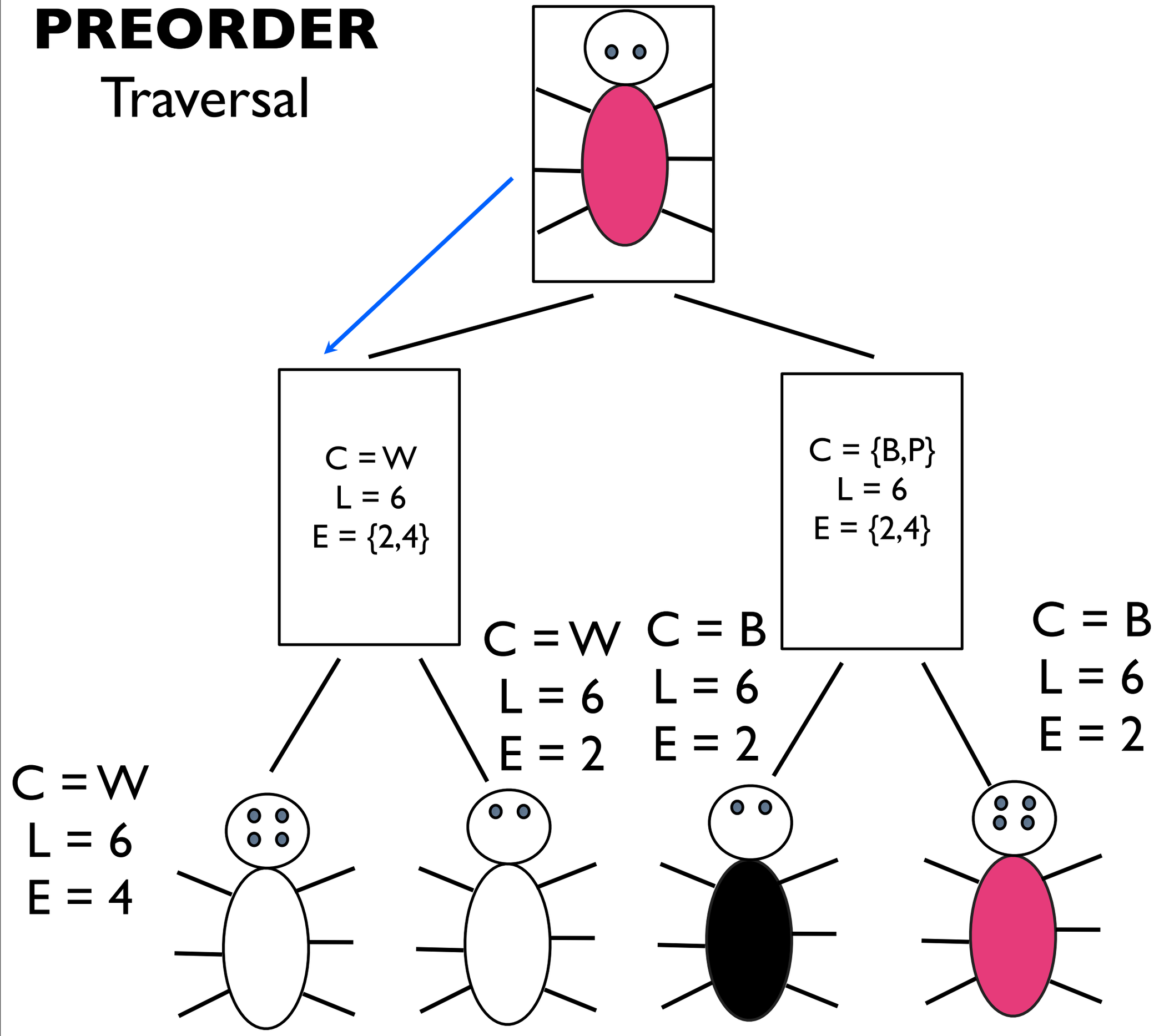
C = B
L = 6
E = 2

C = P
L = 6
E = 4



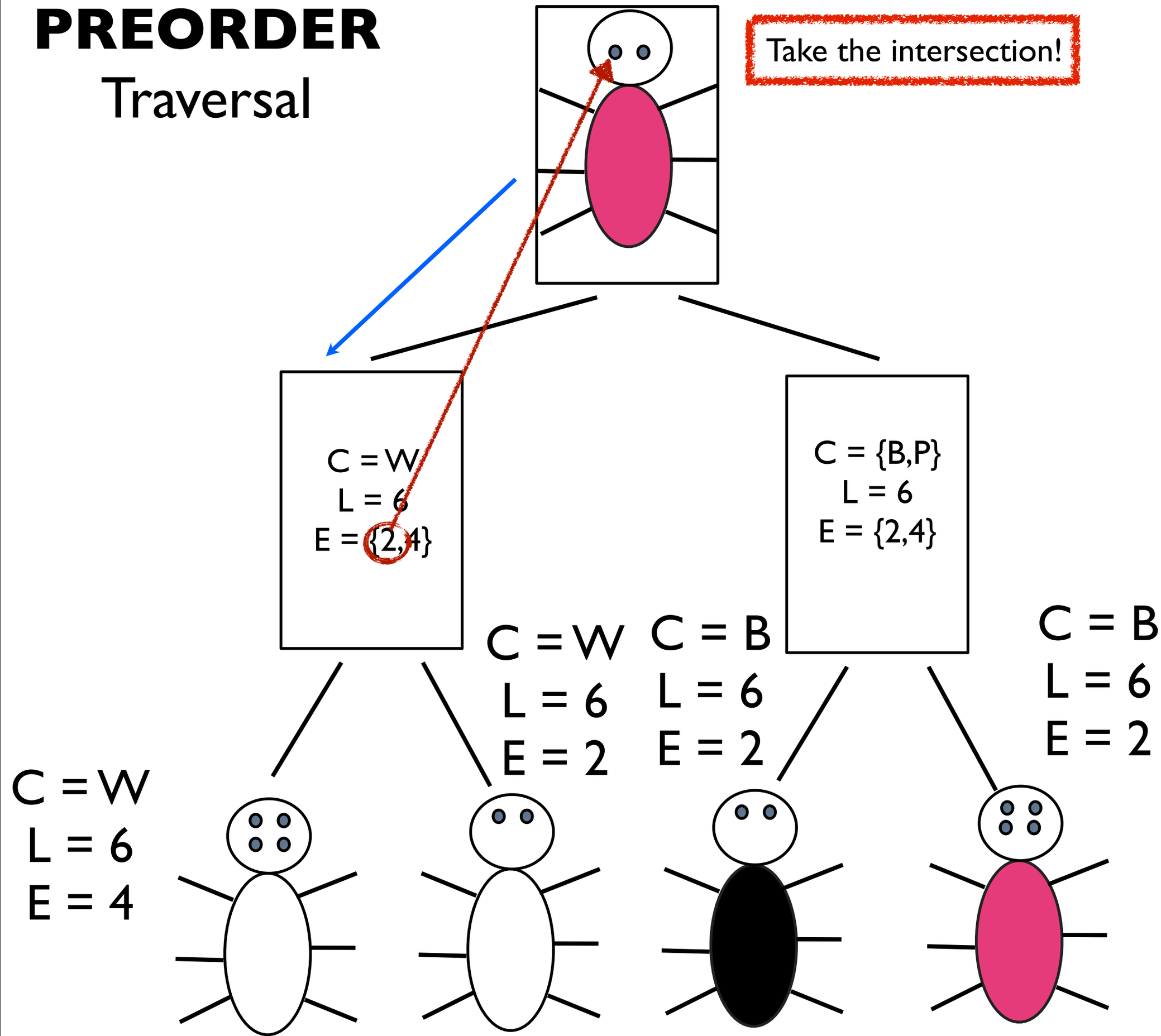
PREORDER

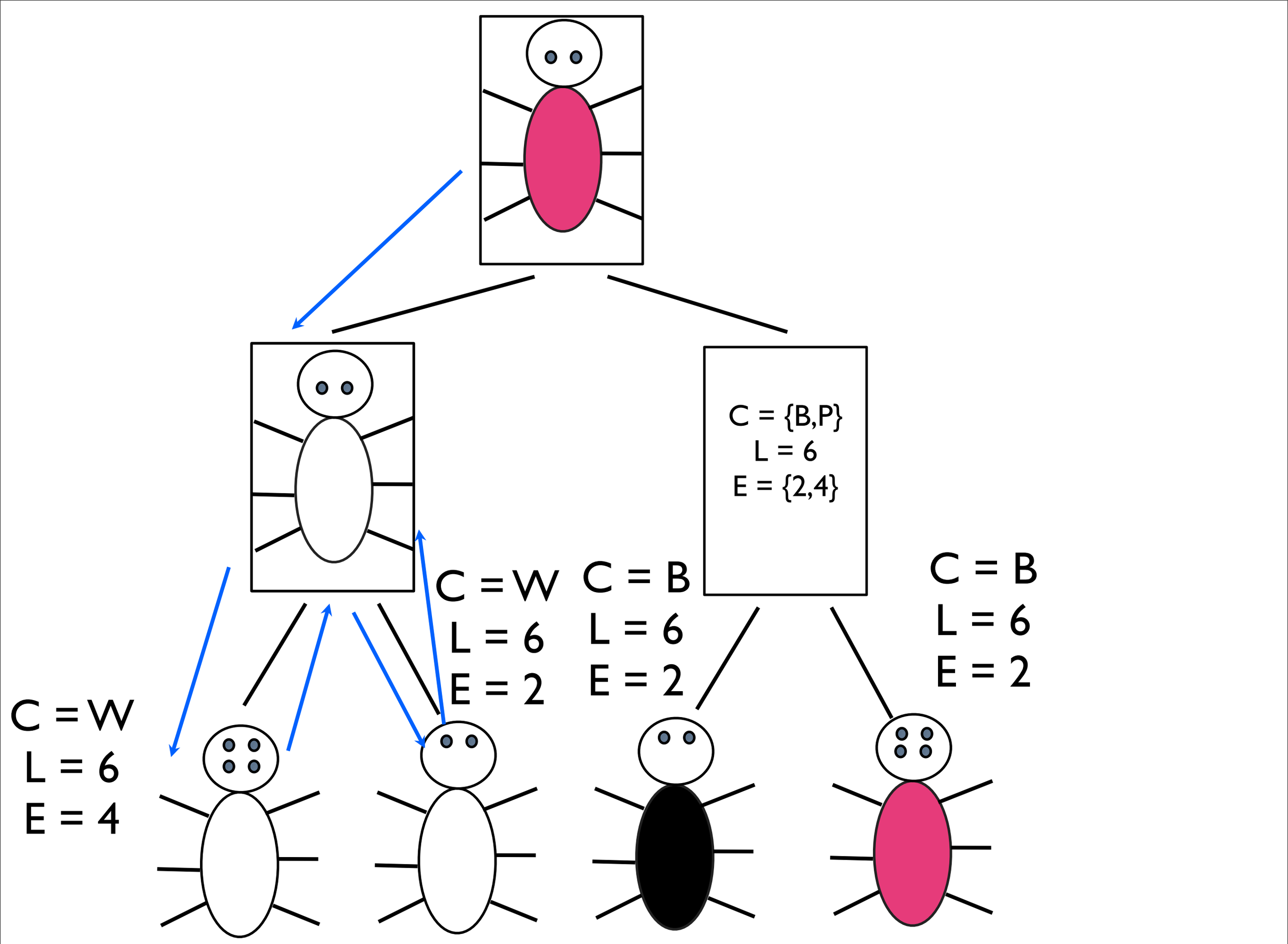
Traversal

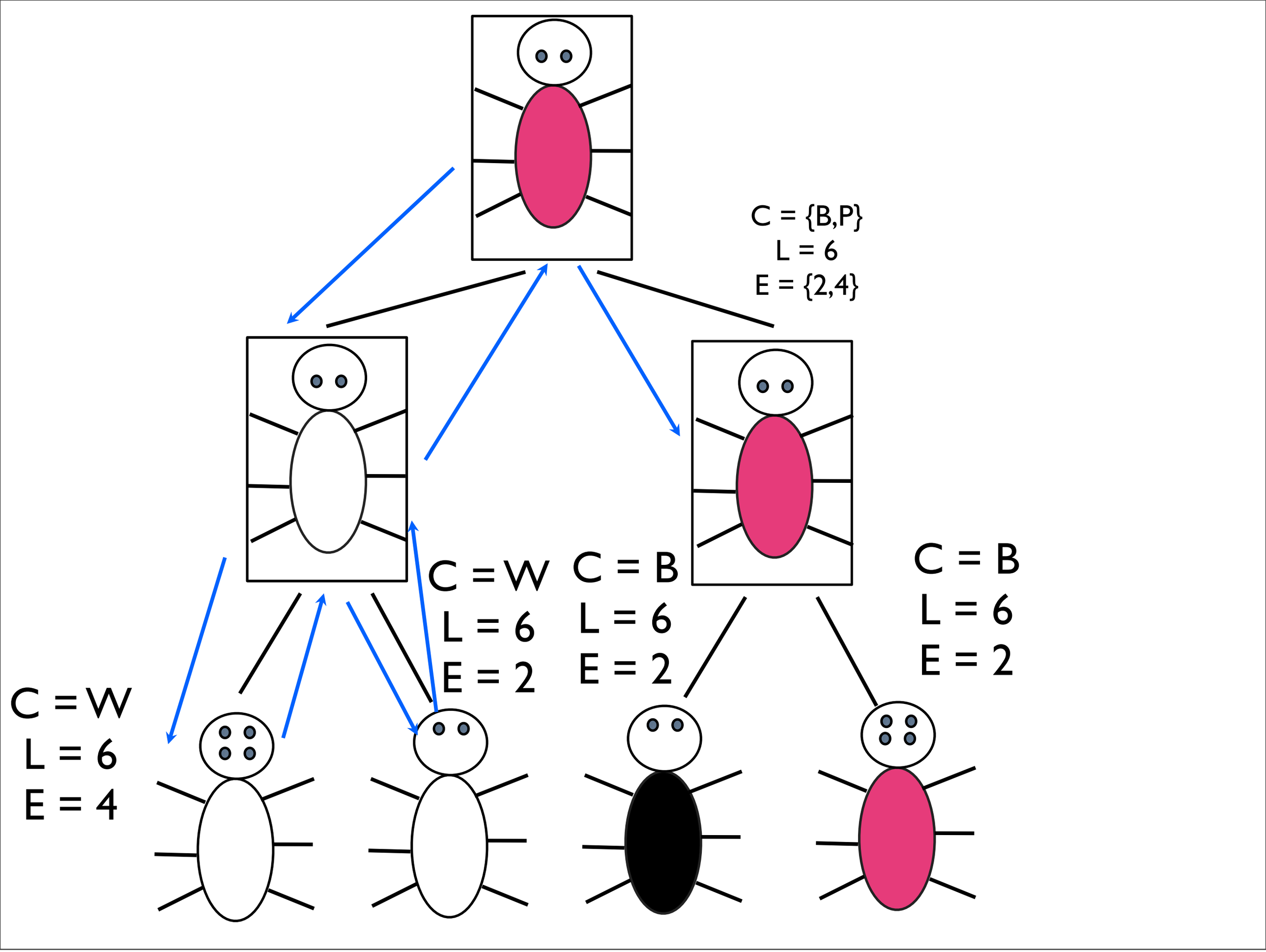


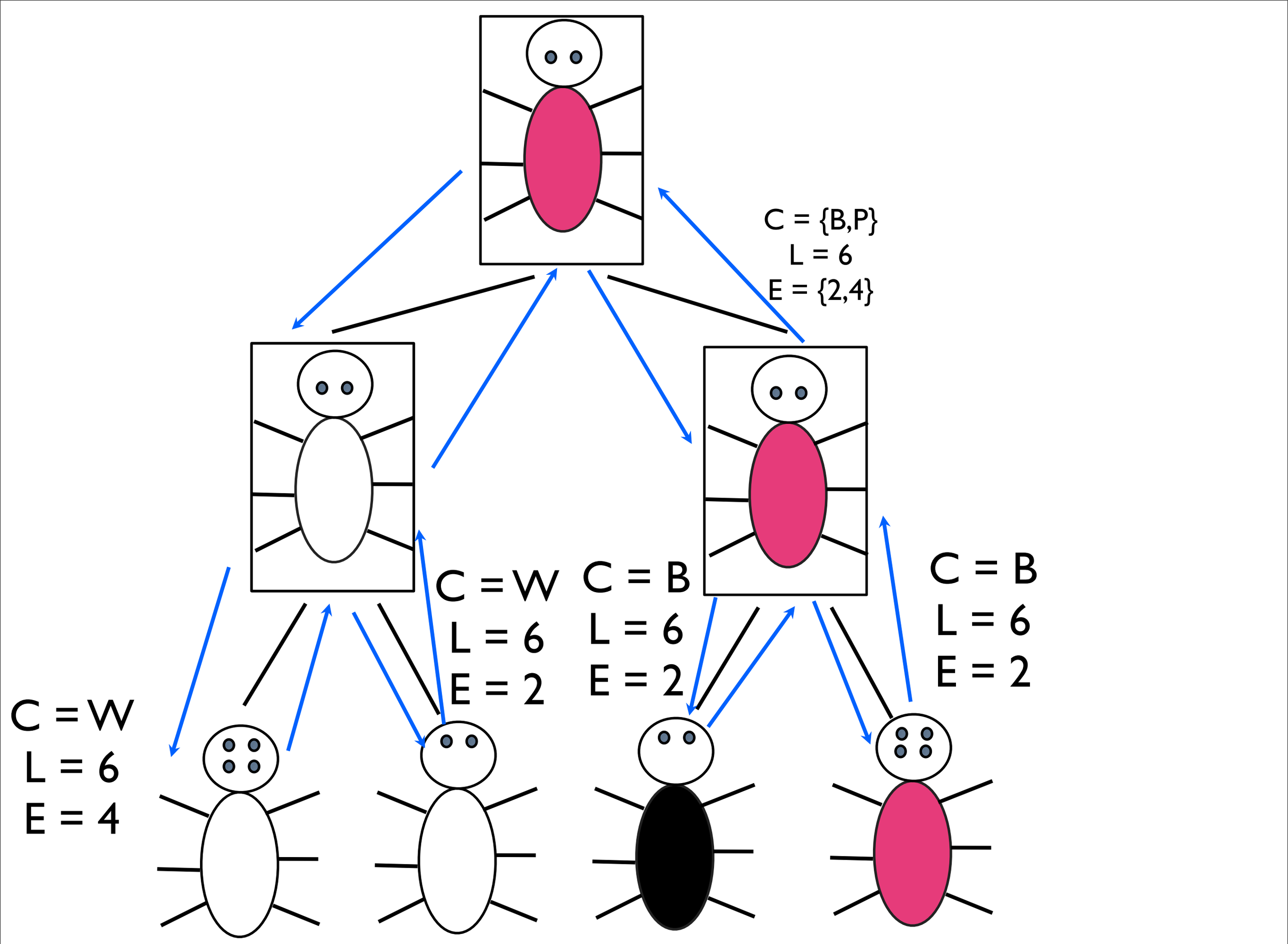
PREORDER

Traversal

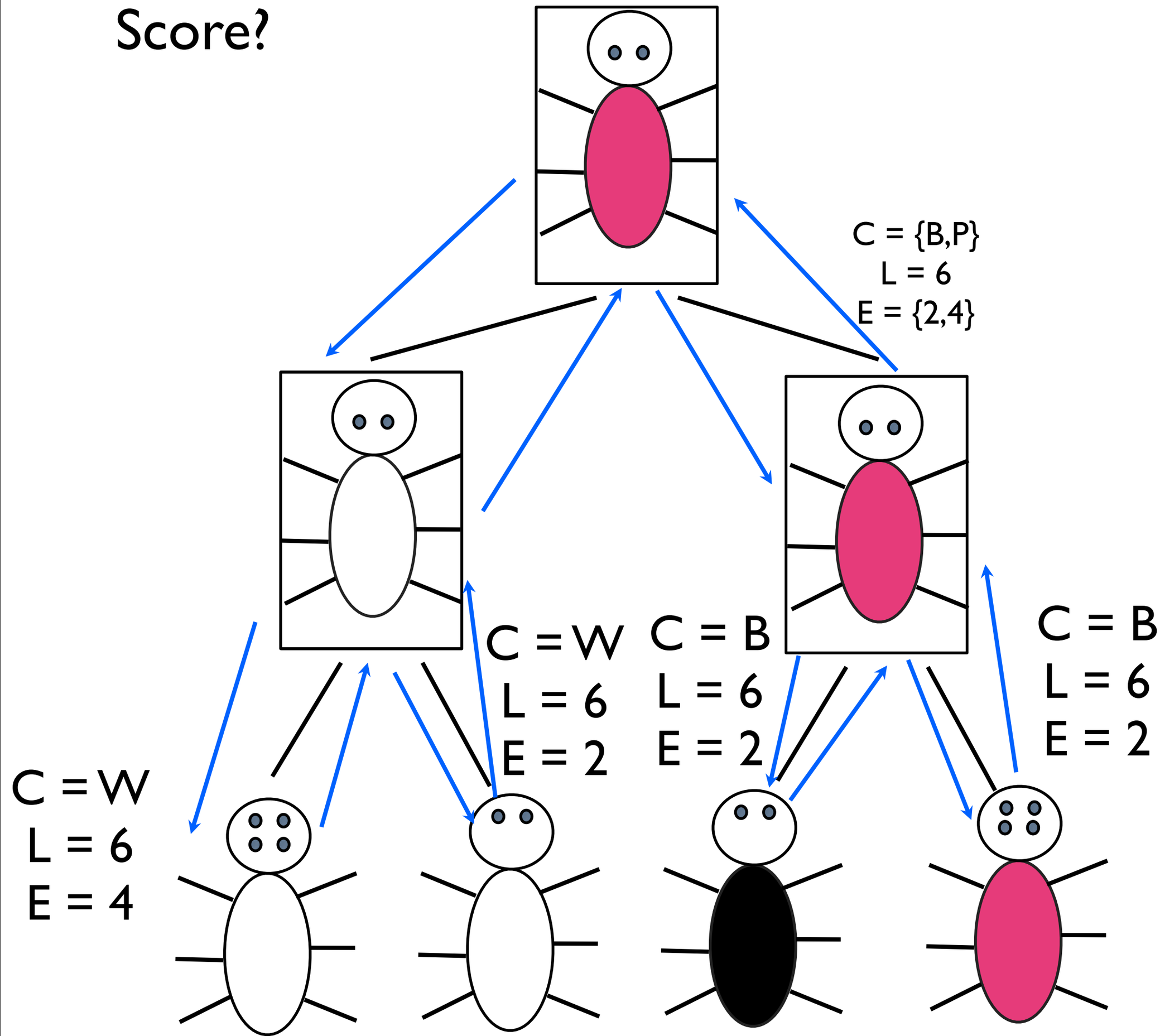


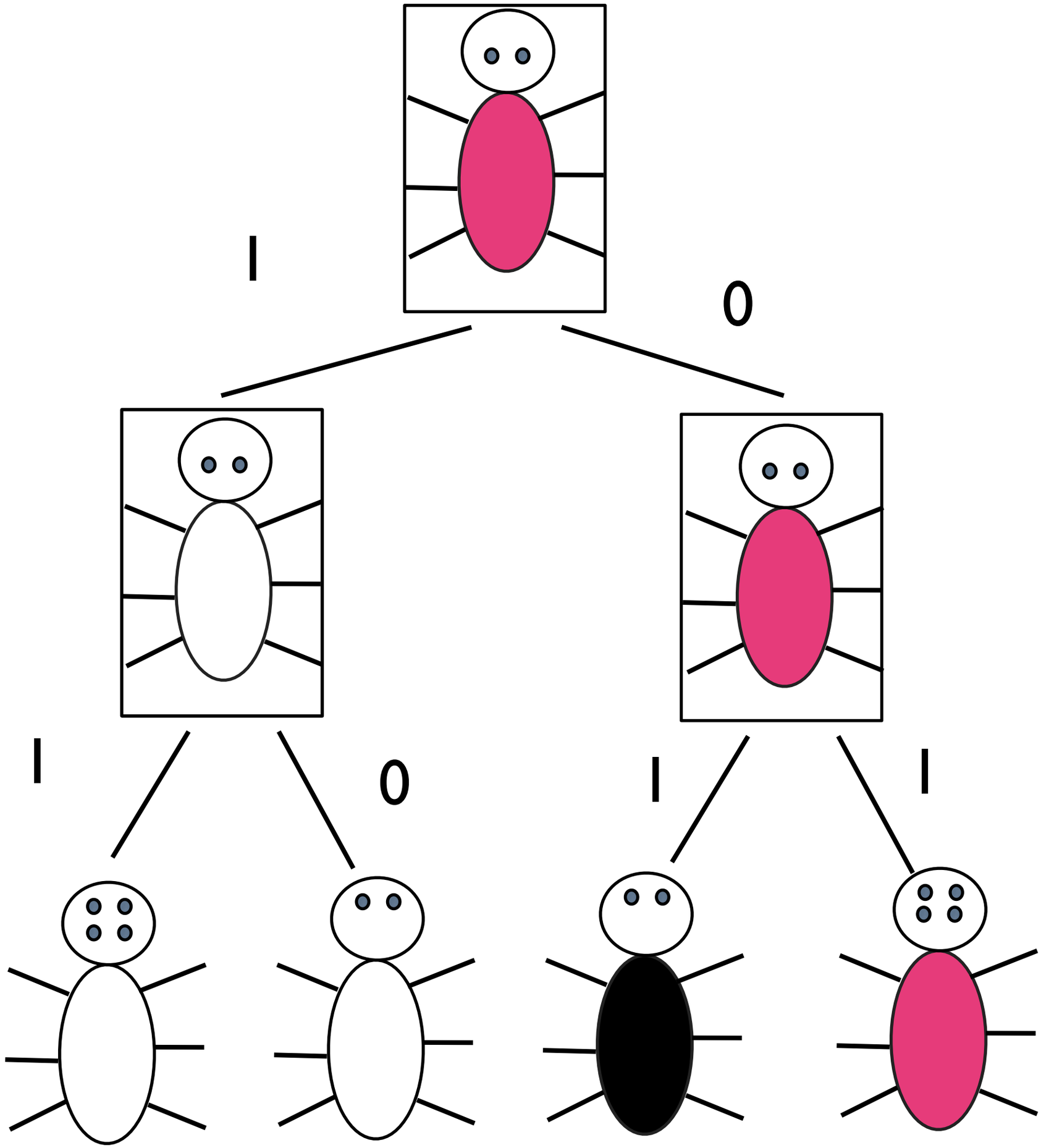

C = {B,P}
L = 6
E = {2,4}
C = W
L = 6
E = 4
C = W
L = 6
E = 2
C = B
L = 6
E = 2
C = B
L = 6
E = 2


C = {B,P}
L = 6
E = {2,4}
C = W
L = 6
E = 4
C = W
L = 6
E = 2
C = B
L = 6
E = 2
C = B
L = 6
E = 2

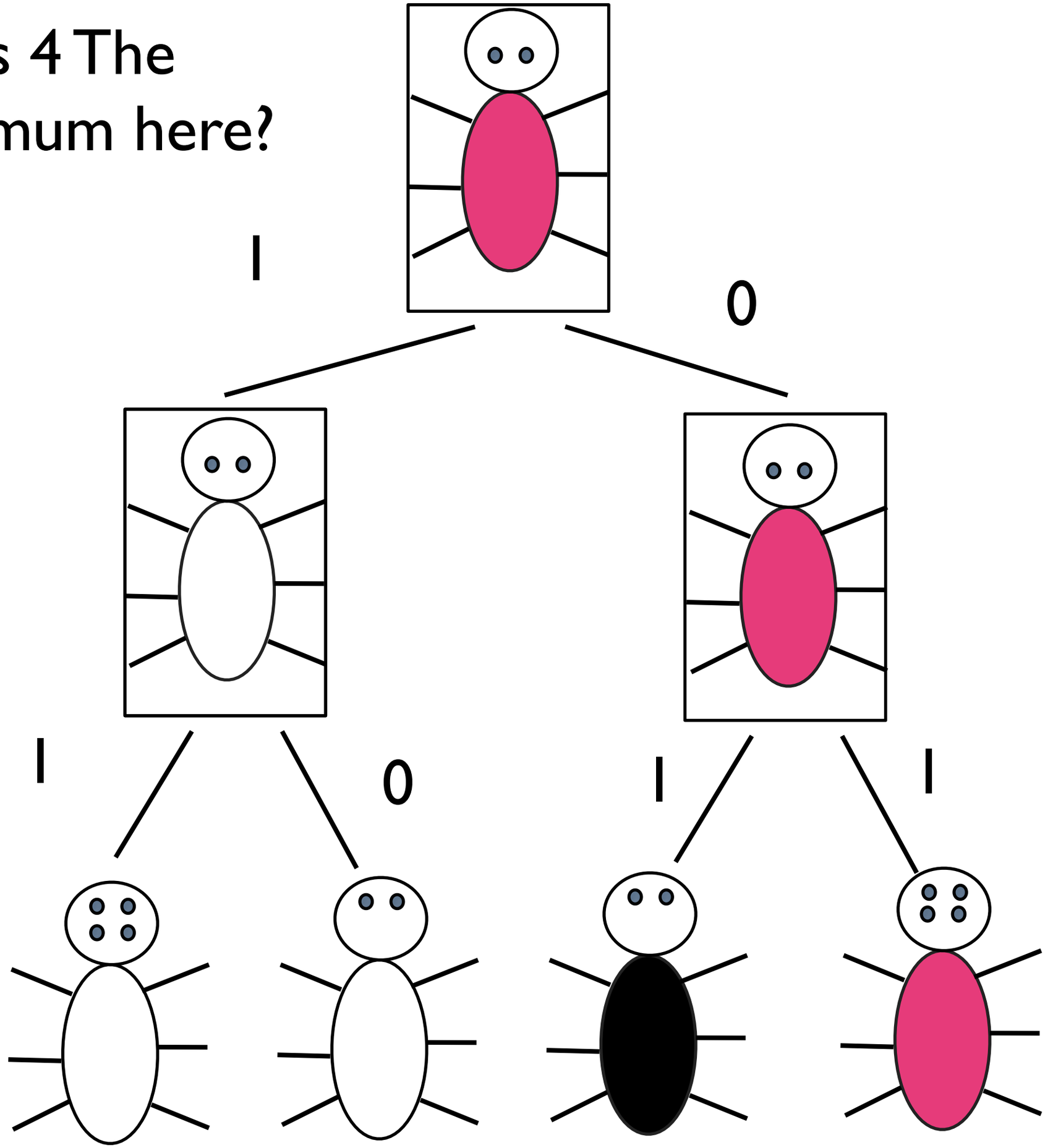

C = {B,P}
L = 6
E = {2,4}
C = W
L = 6
E = 4
C = W
L = 6
E = 2
C = B
L = 6
E = 2
C = B
L = 6
E = 2

Score?

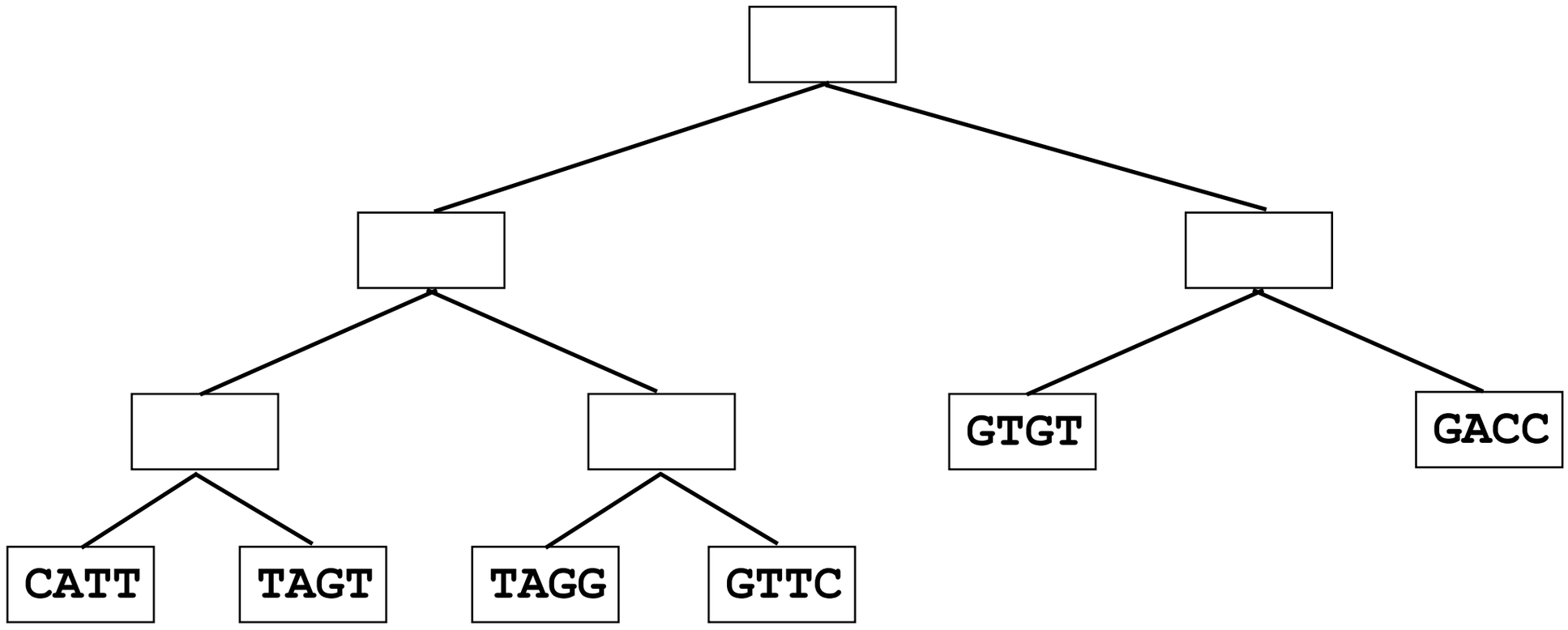



1
0
1
0
1
1

Is 4 The minimum here?

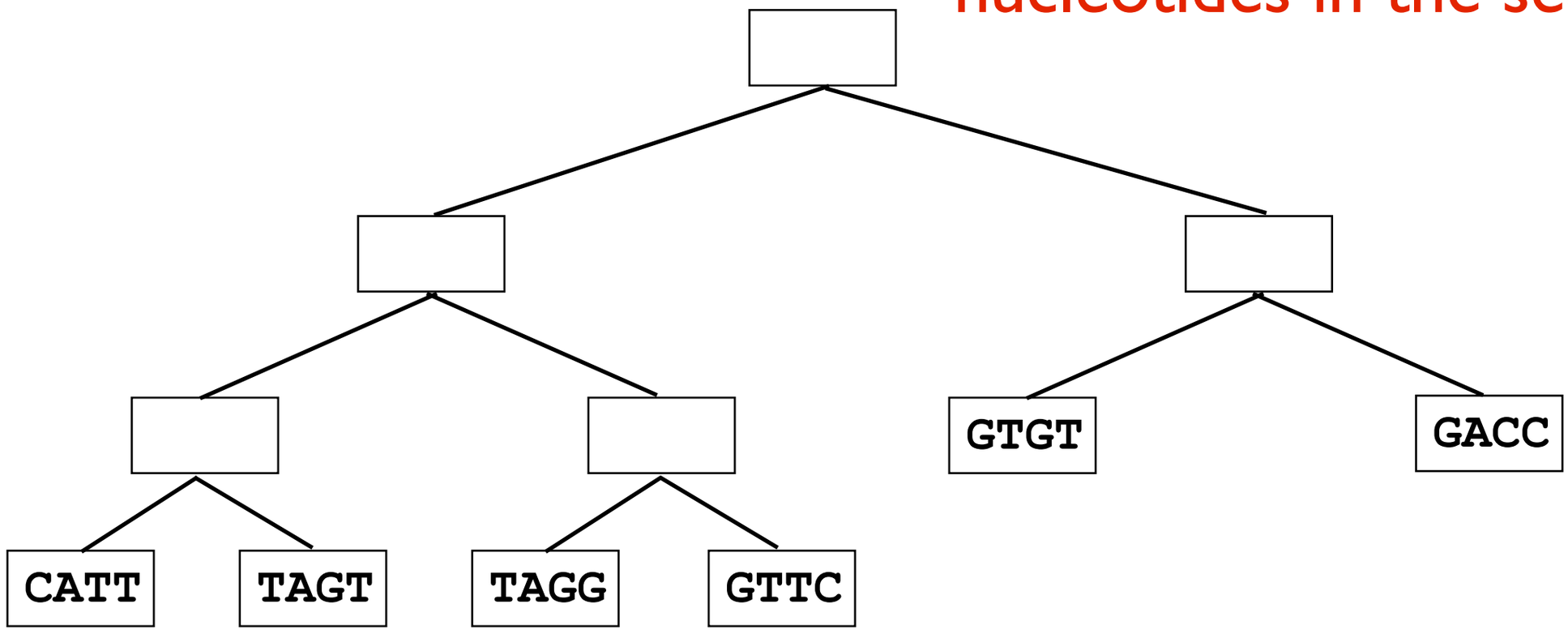


Fitch's on DNA

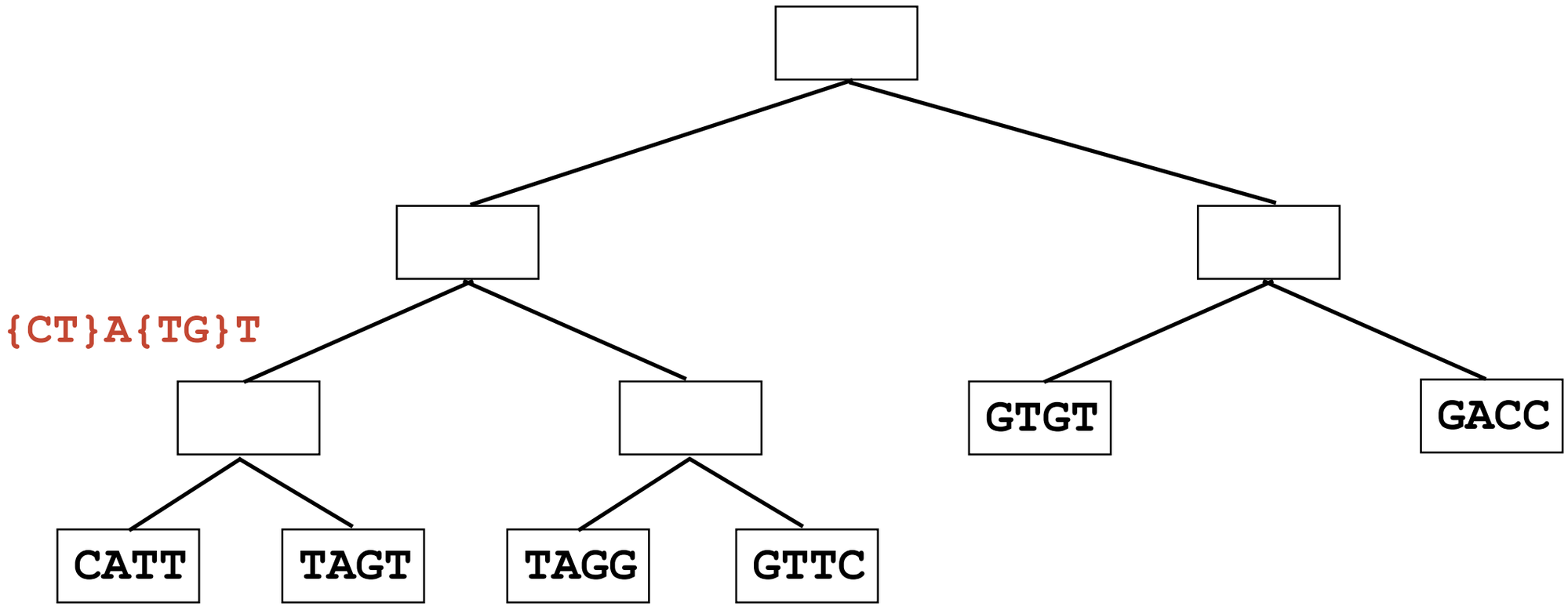


Fitch's Algorithm on DNA

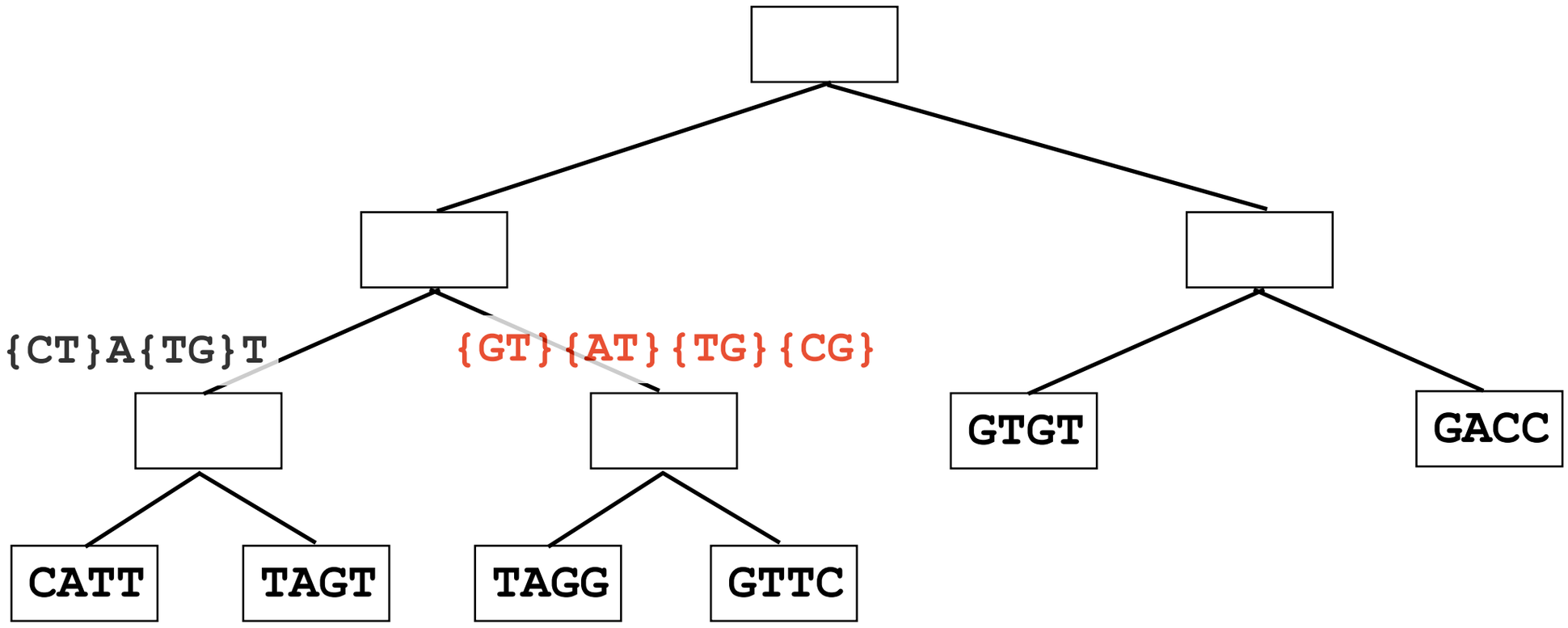
Features are the 1, 2, 3, 4th nucleotides in the sequence



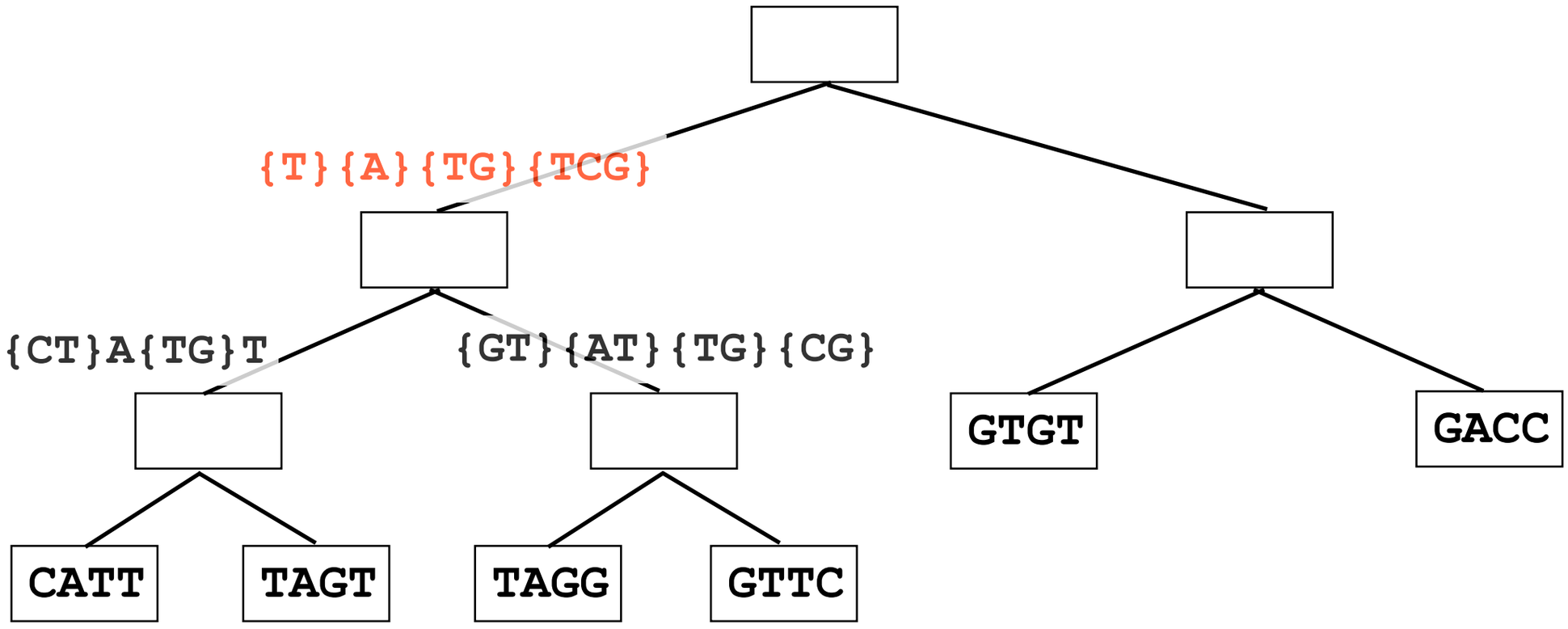
Fitch's on DNA



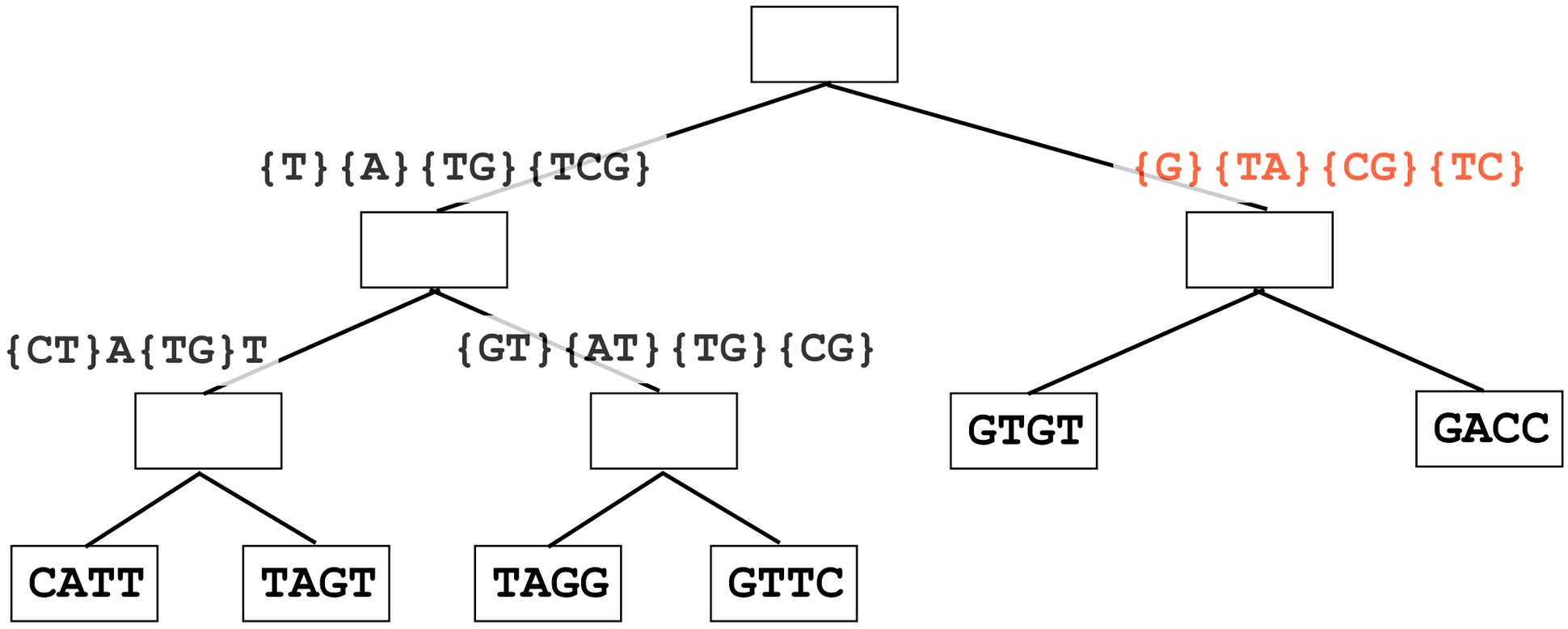
Fitch's on DNA



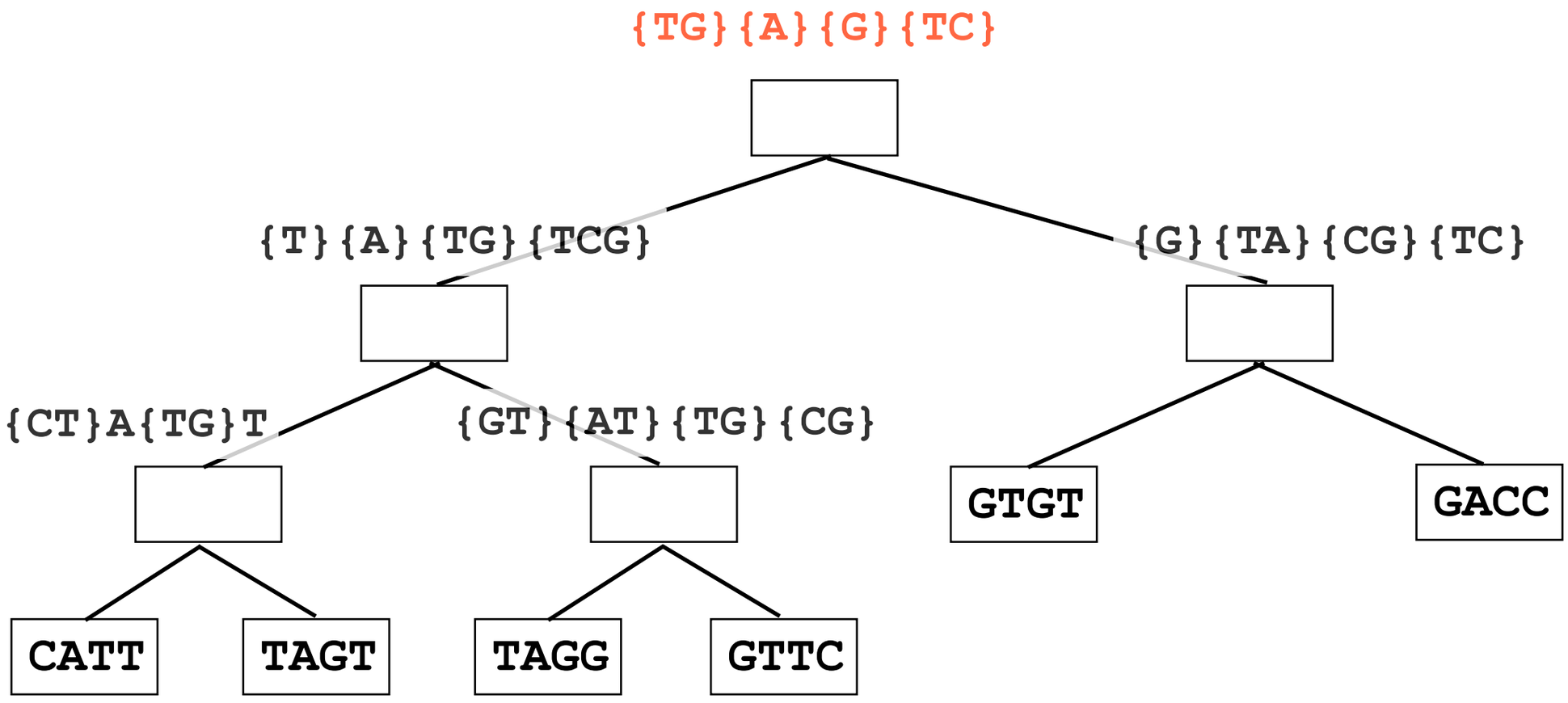
Fitch's on DNA



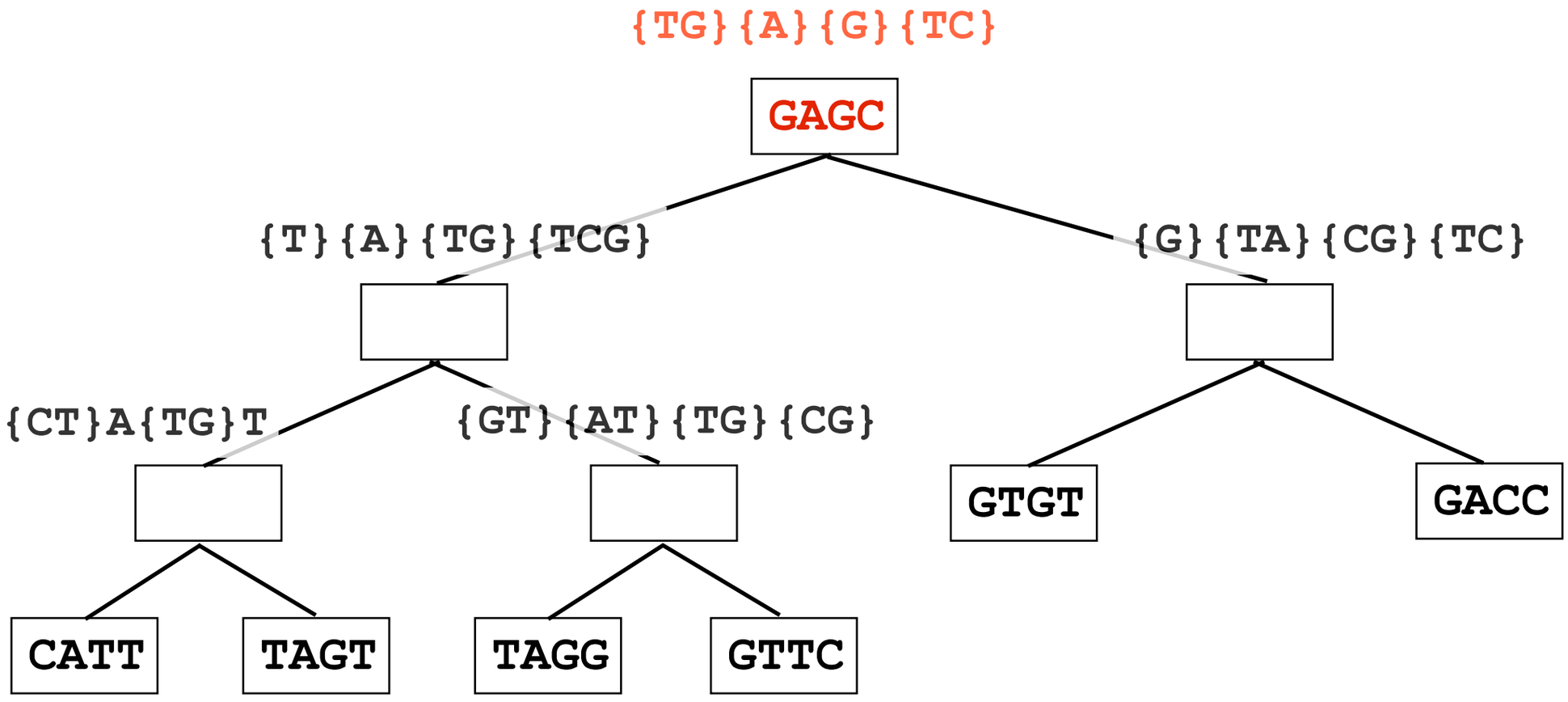
Fitch's on DNA



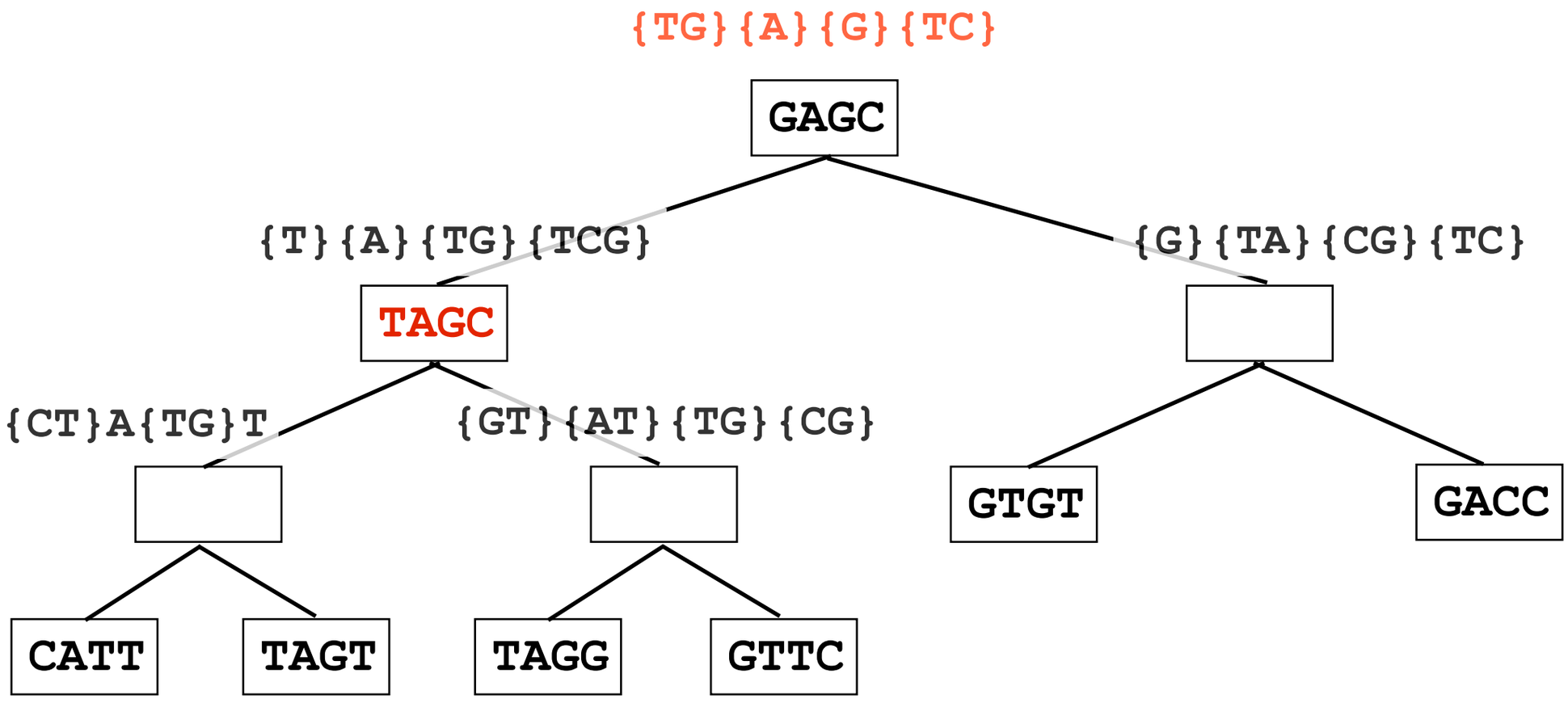
Fitch's on DNA



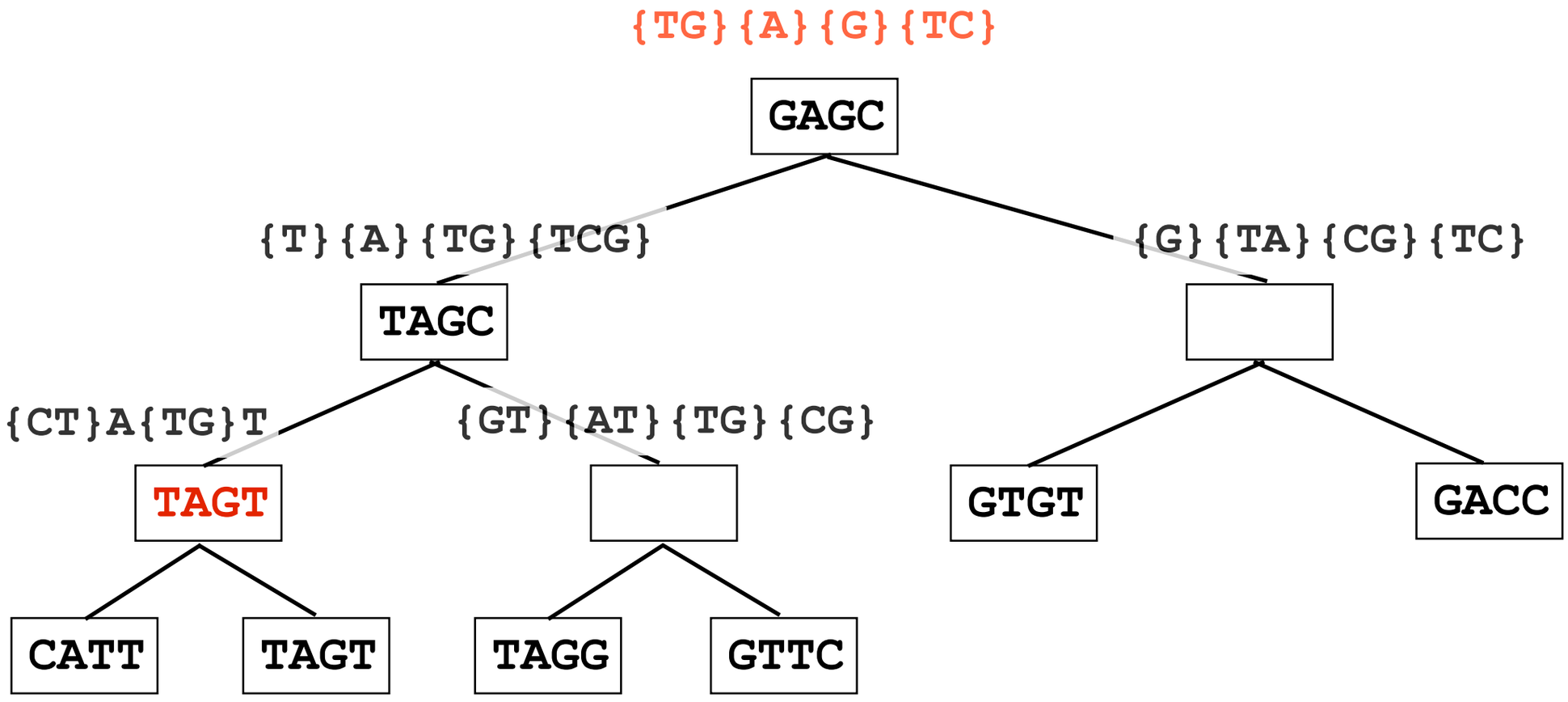
Fitch's on DNA



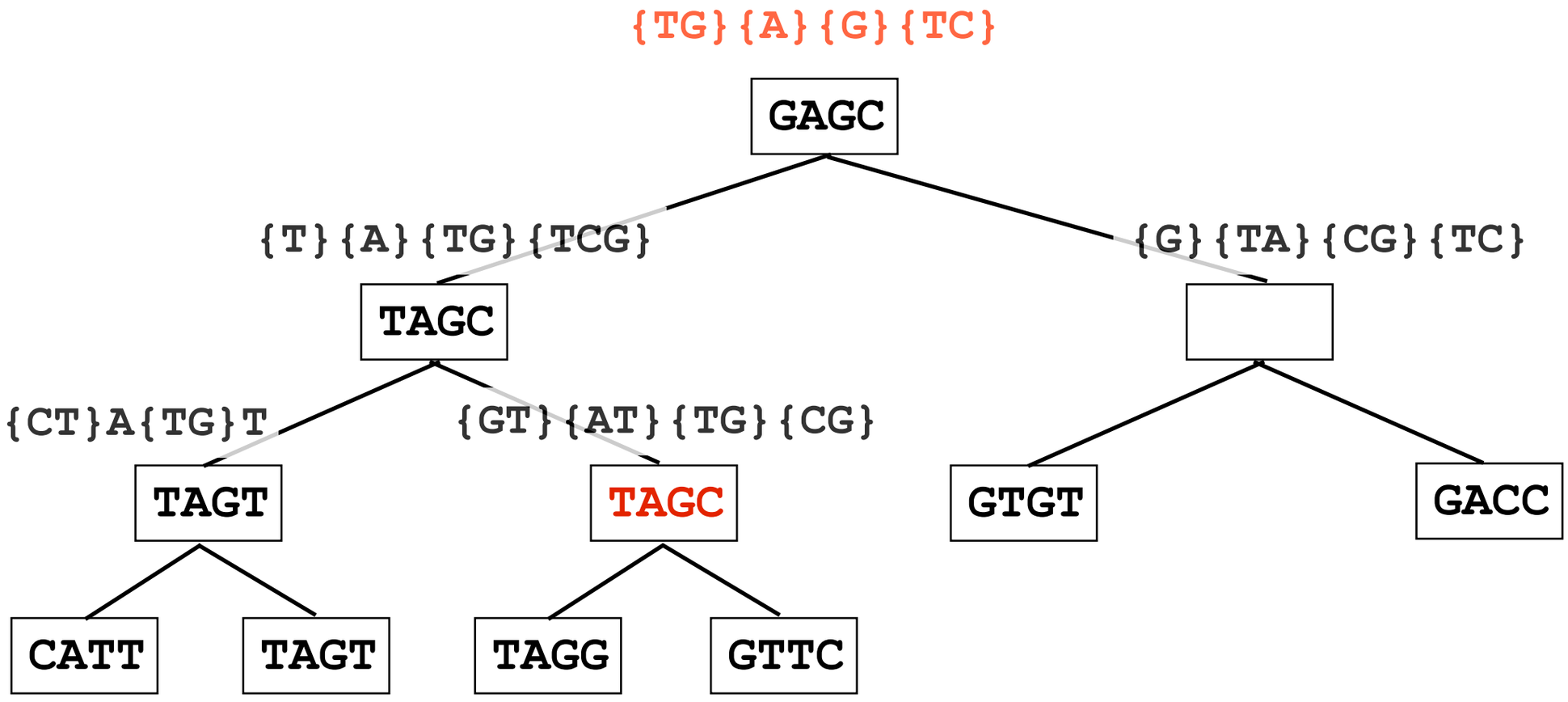
Fitch's on DNA



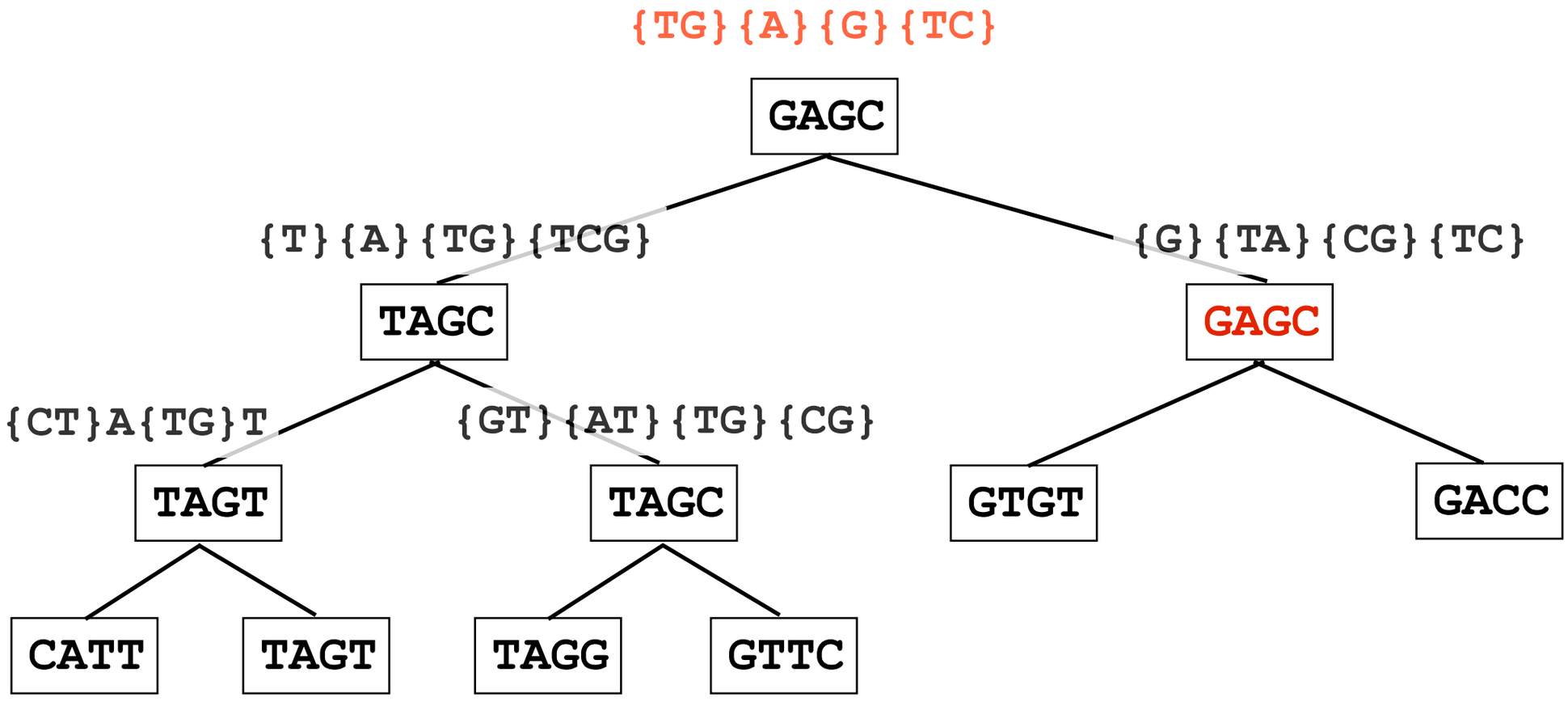
Fitch's on DNA



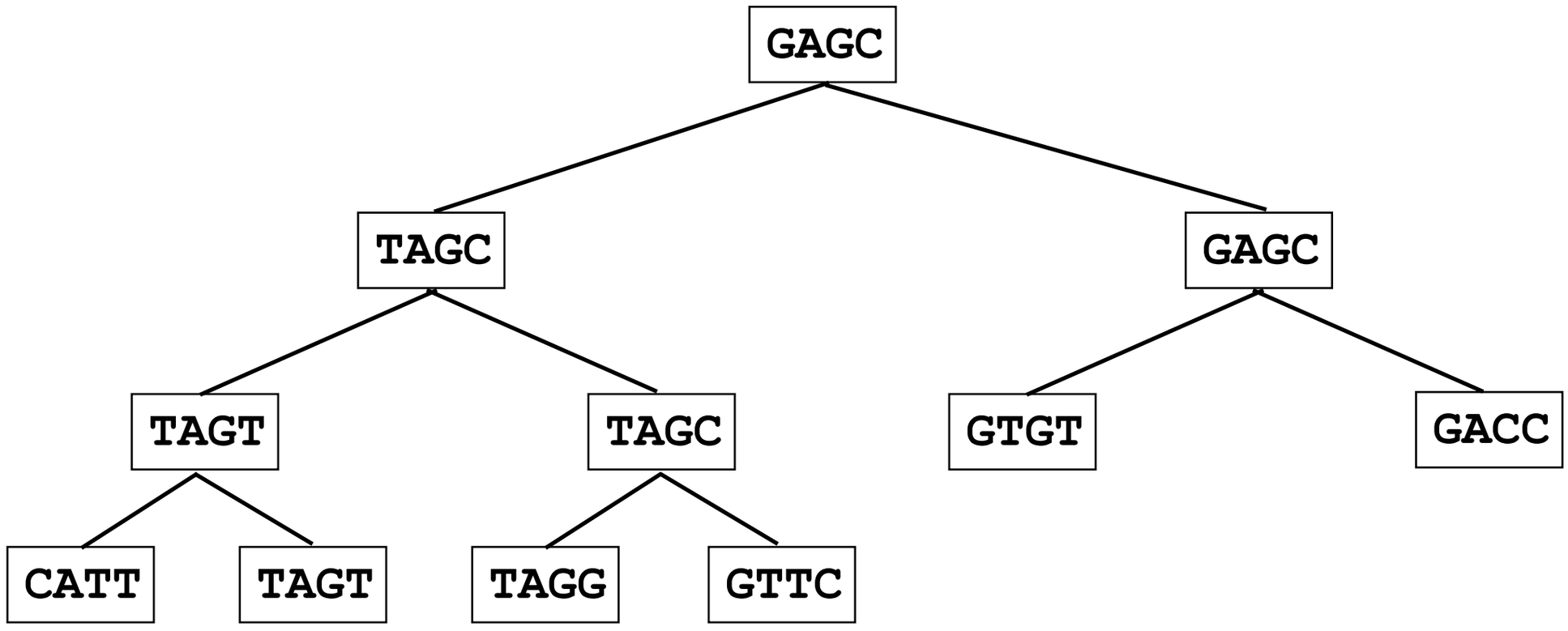
Fitch's on DNA



Fitch's on DNA

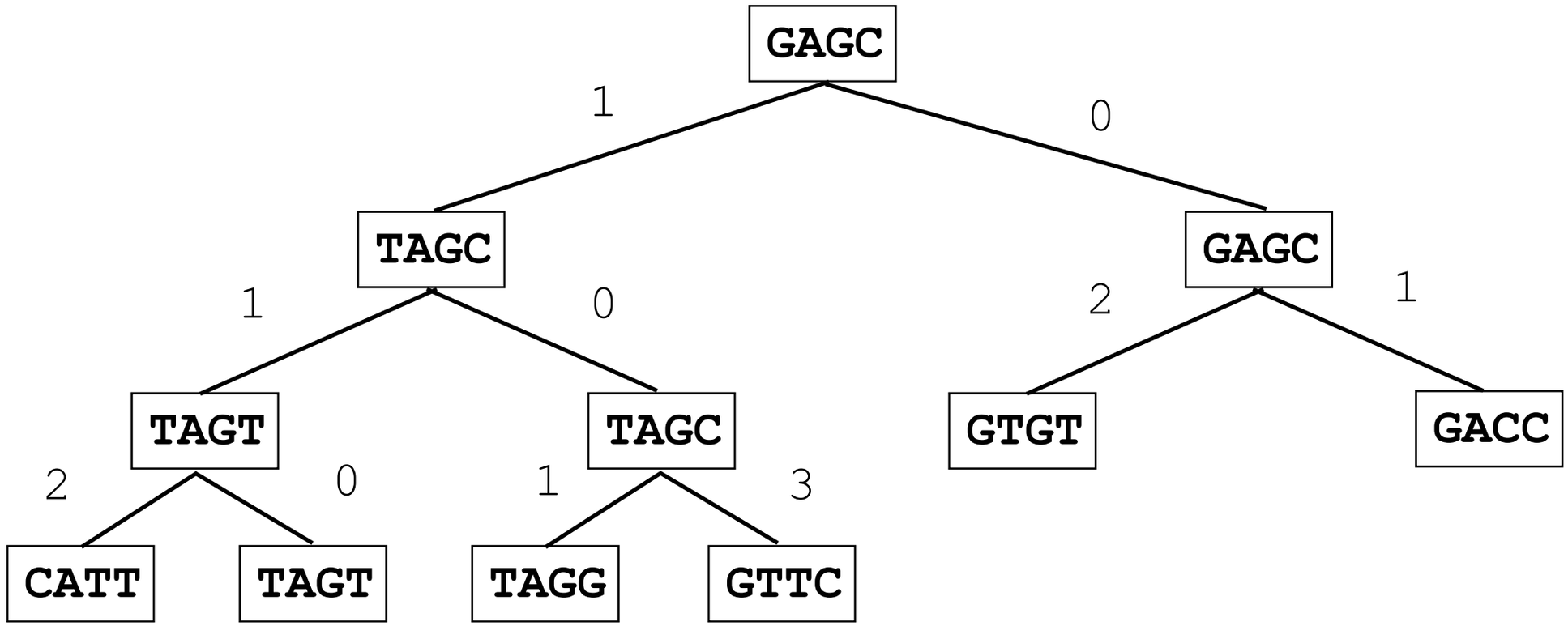


Fitch's on DNA



What is parsimony of this tree?

Fitch's on DNA



The parsimony of this tree is: 11