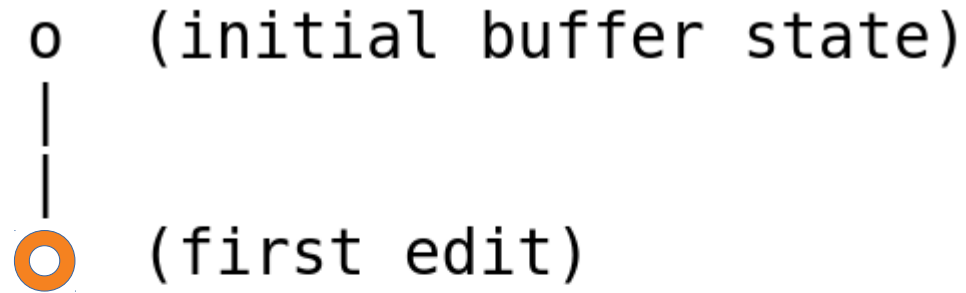


# Undo systems

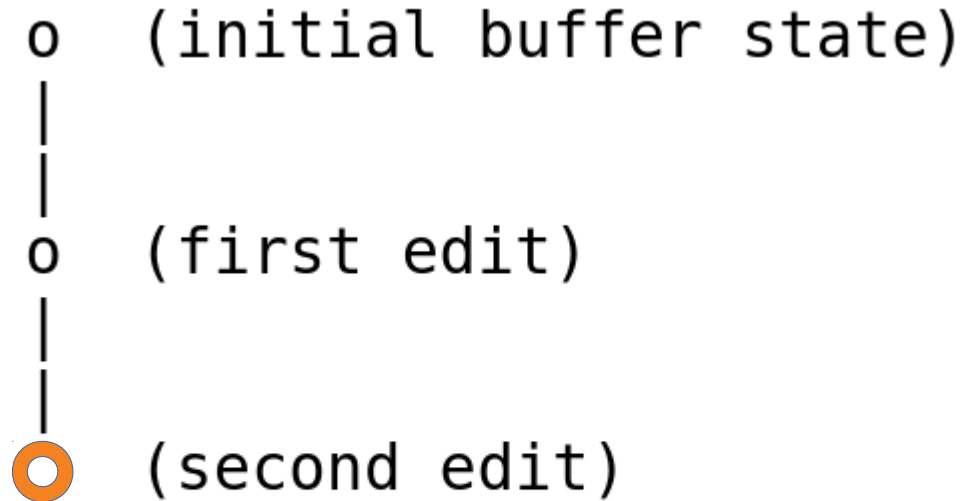
# Linear systems \_ classic (undo/redo)

○ (initial buffer state)

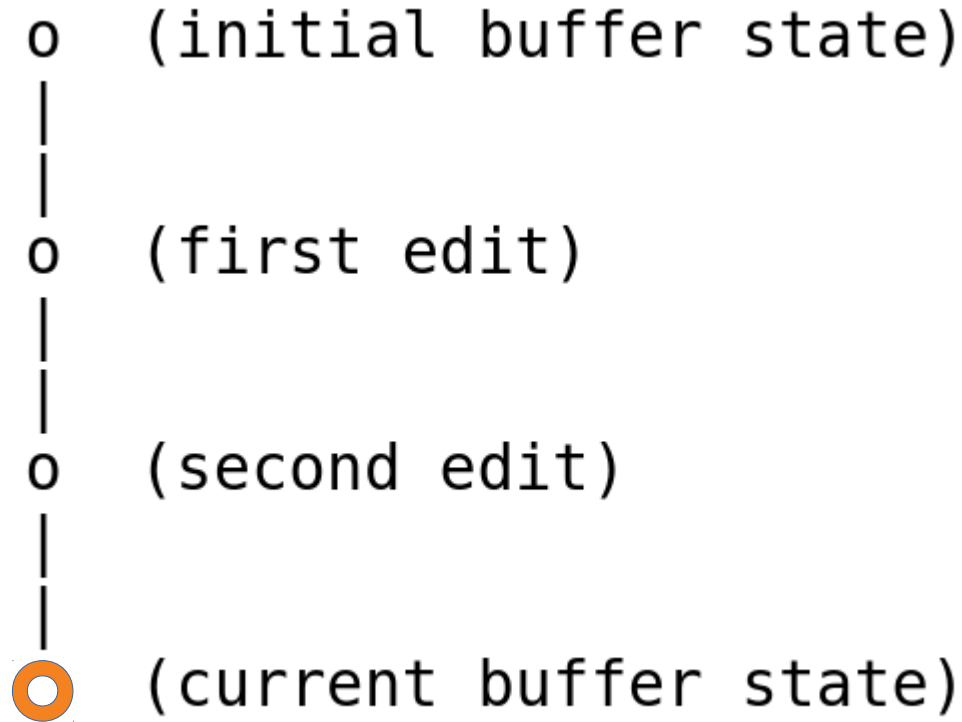
# Linear systems \_ classic (undo/redo)



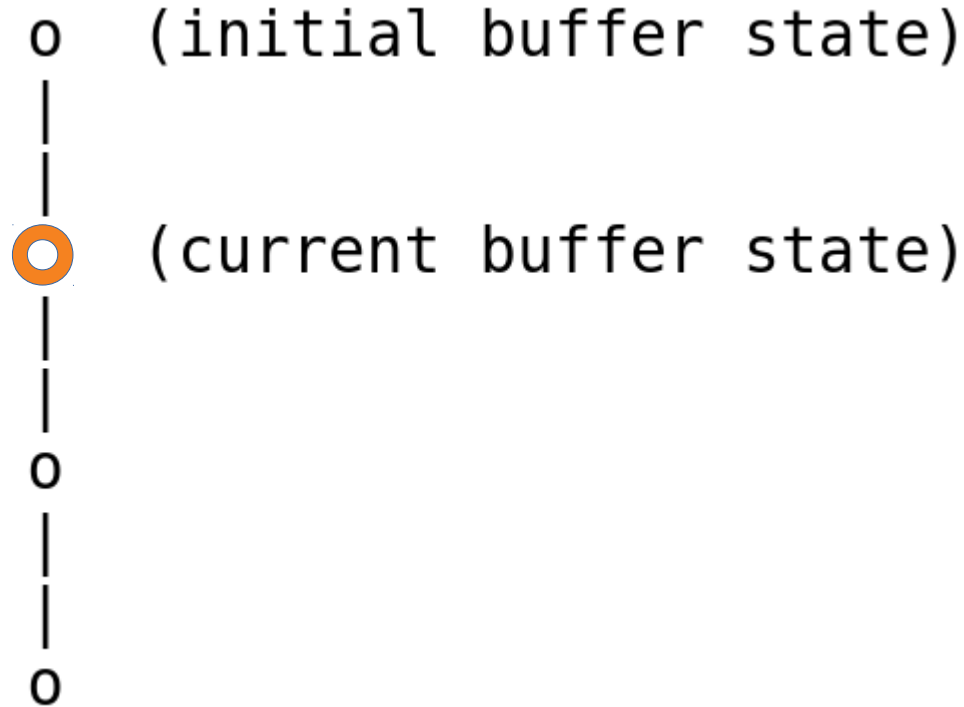
# Linear systems \_ classic (undo/redo)



# Linear systems \_ classic (undo/redo)



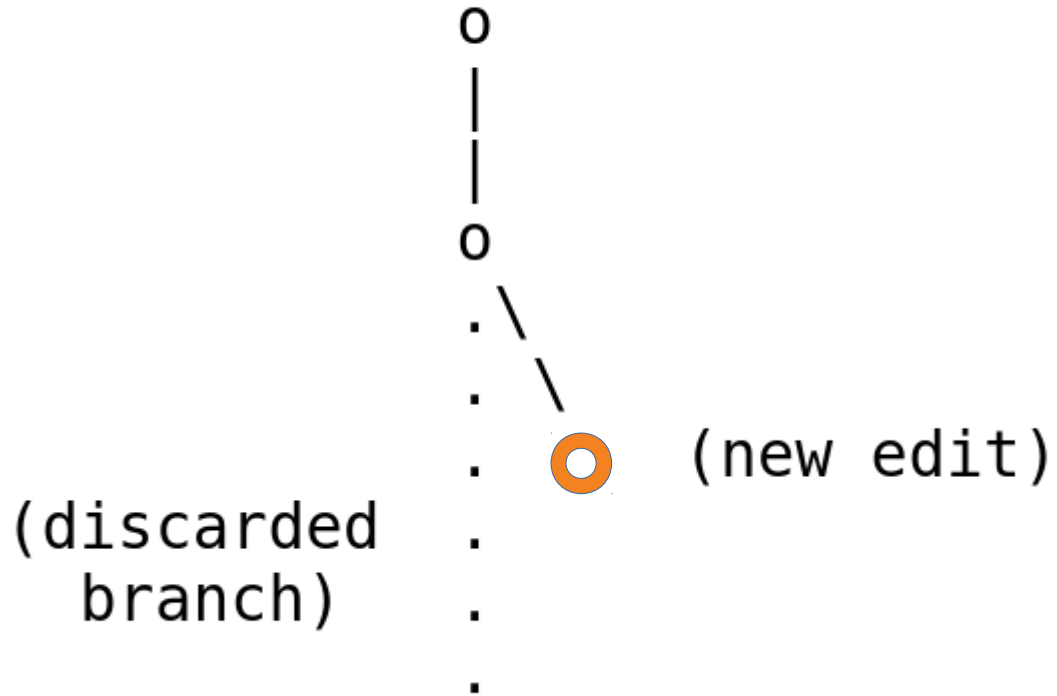
# Linear systems \_ classic (undo/redo)



# Linear systems \_ classic (undo/redo)

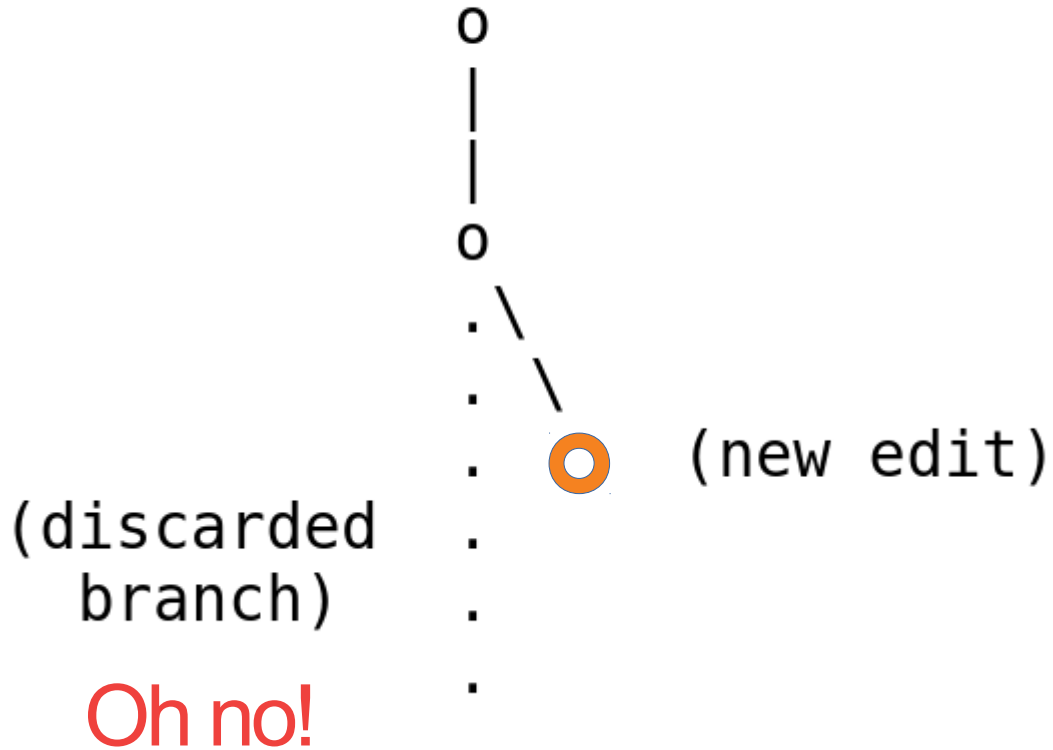


# Linear systems \_ classic (undo/redo)

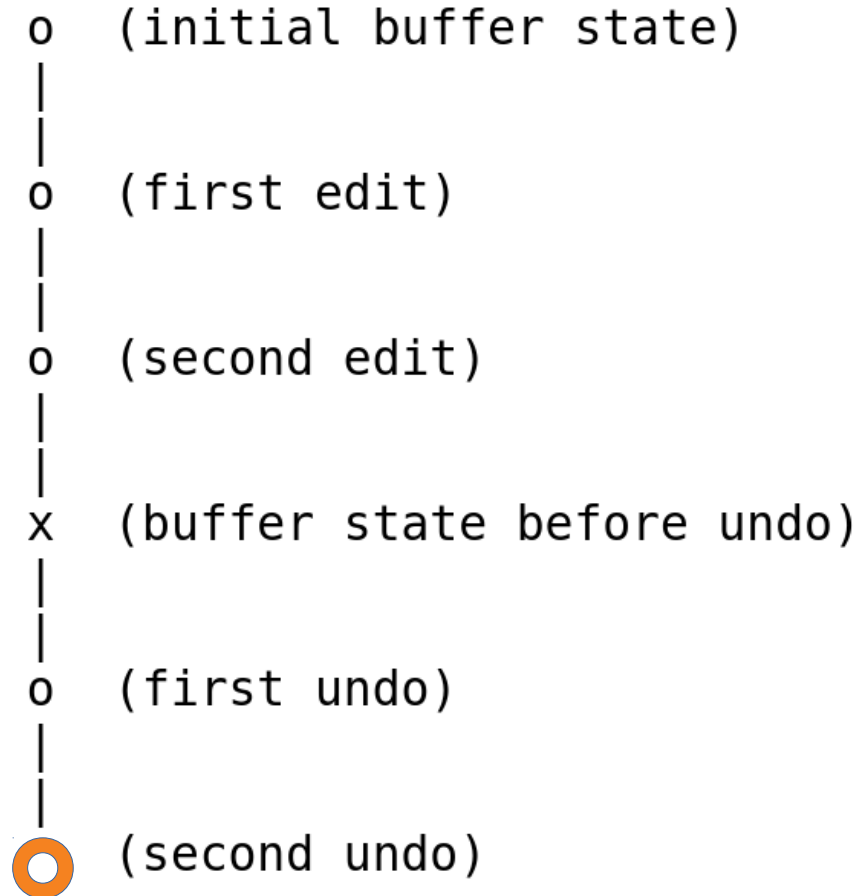




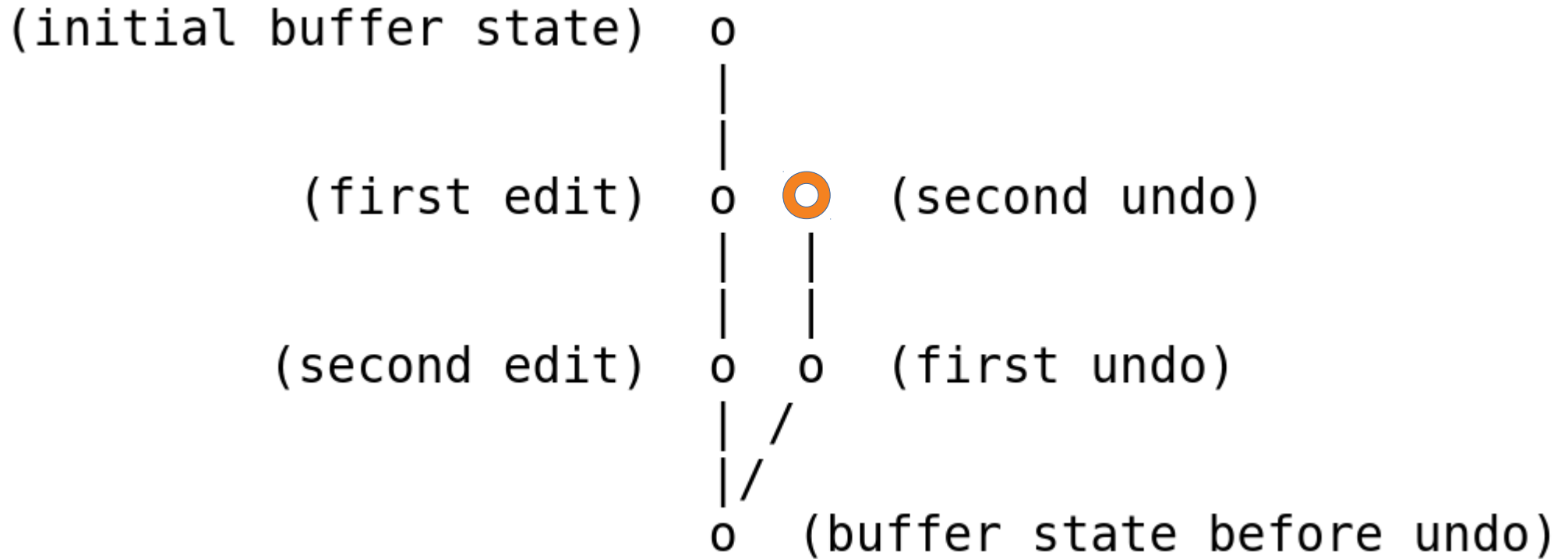
# Linear systems \_ classic (undo/redo)



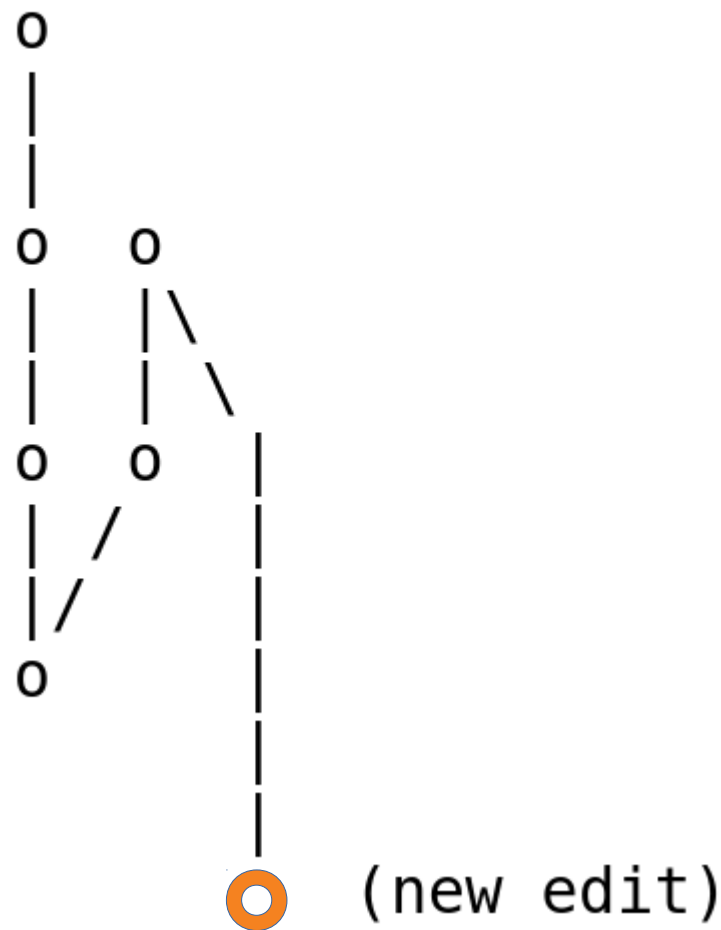
# Linear systems \_ emacs



# Linear systems \_ emacs

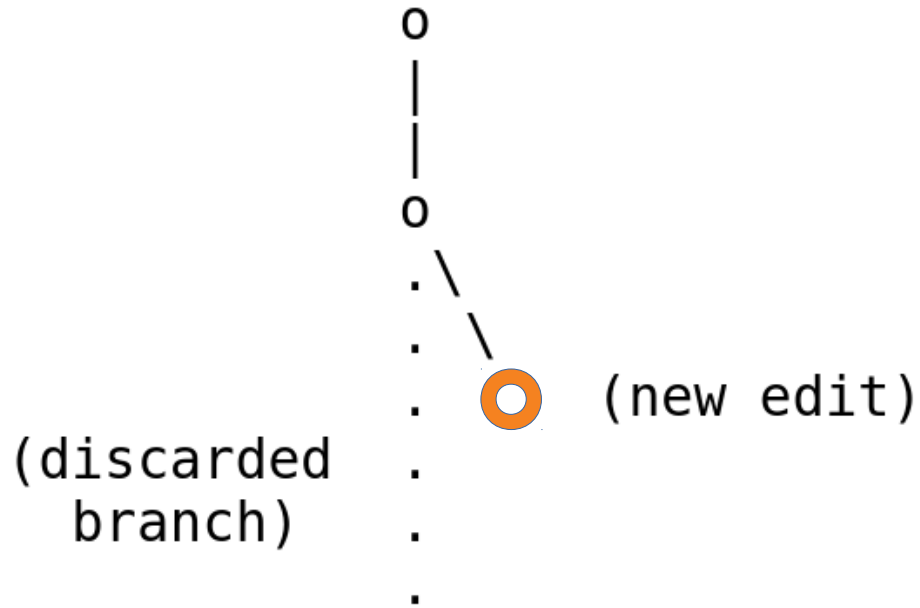


# Linear systems \_ emacs

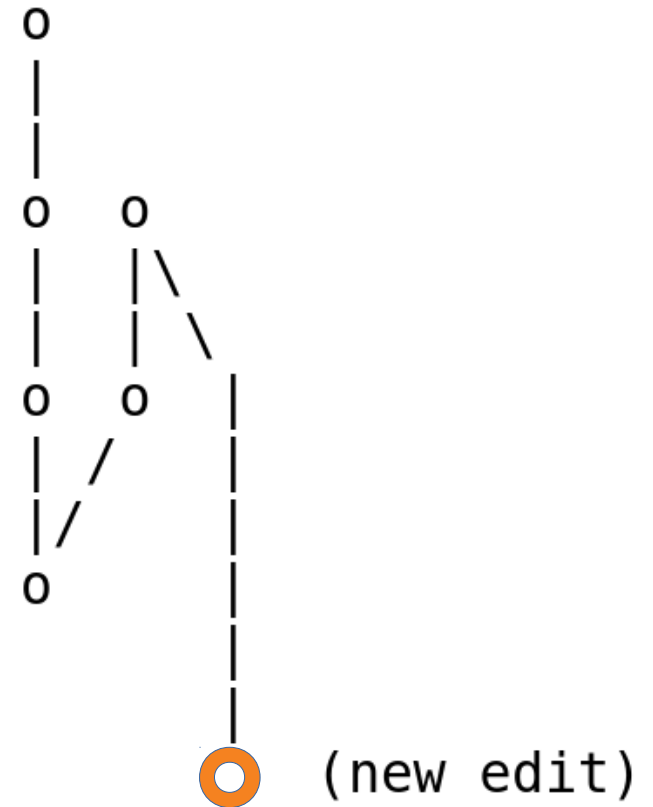


# Linear systems \_ summary

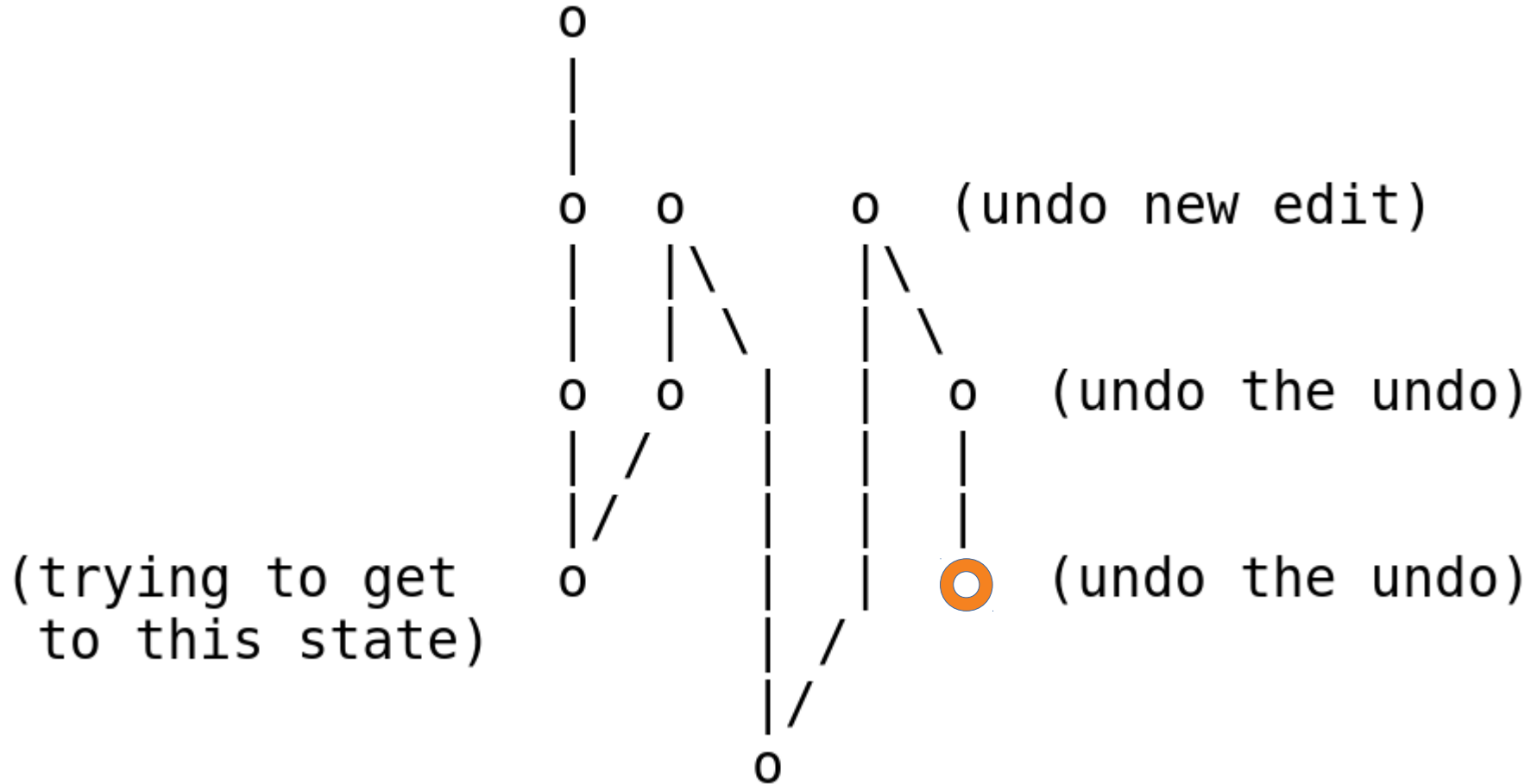
Undo/Redo:



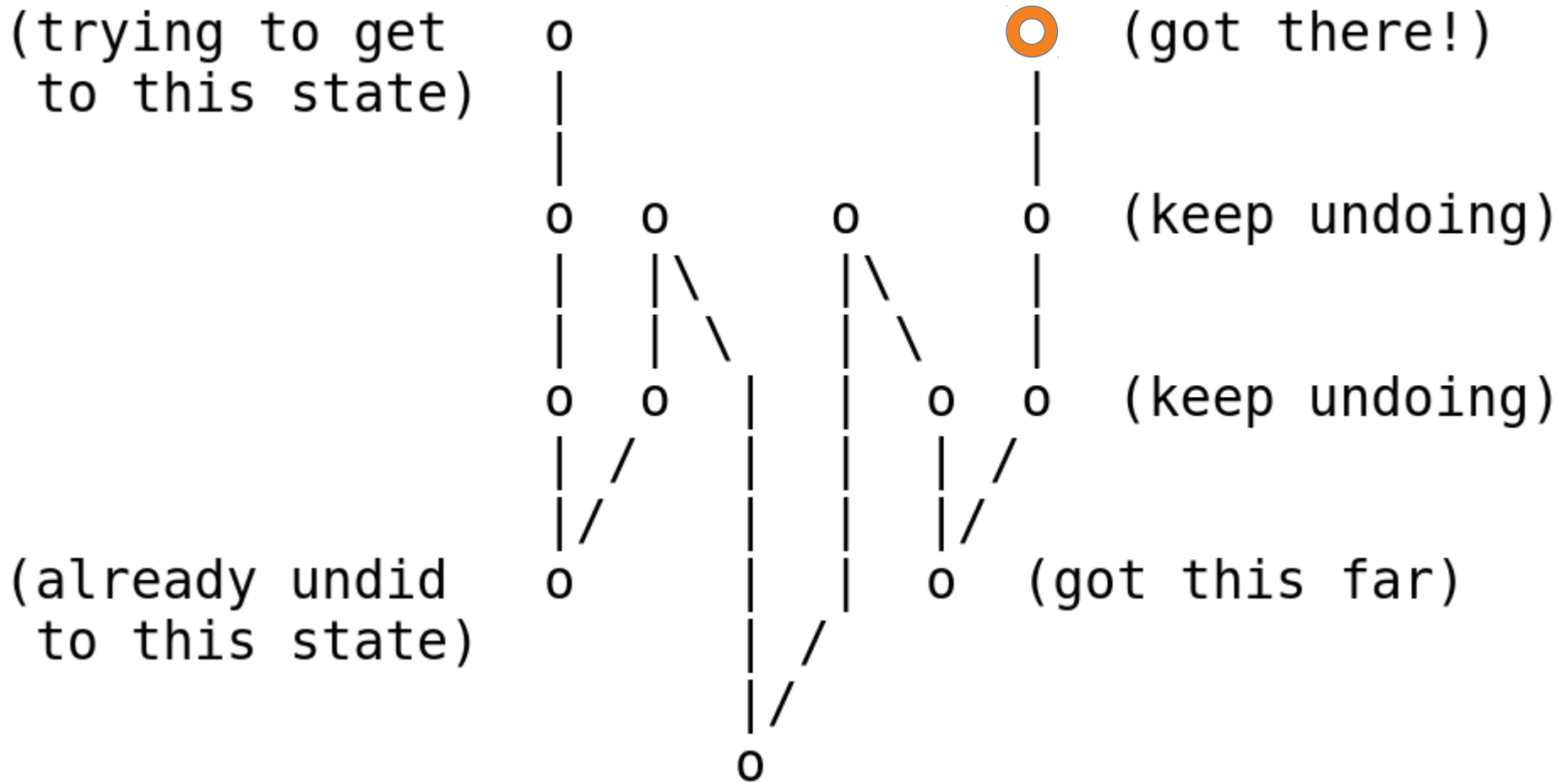
Emacs' undo



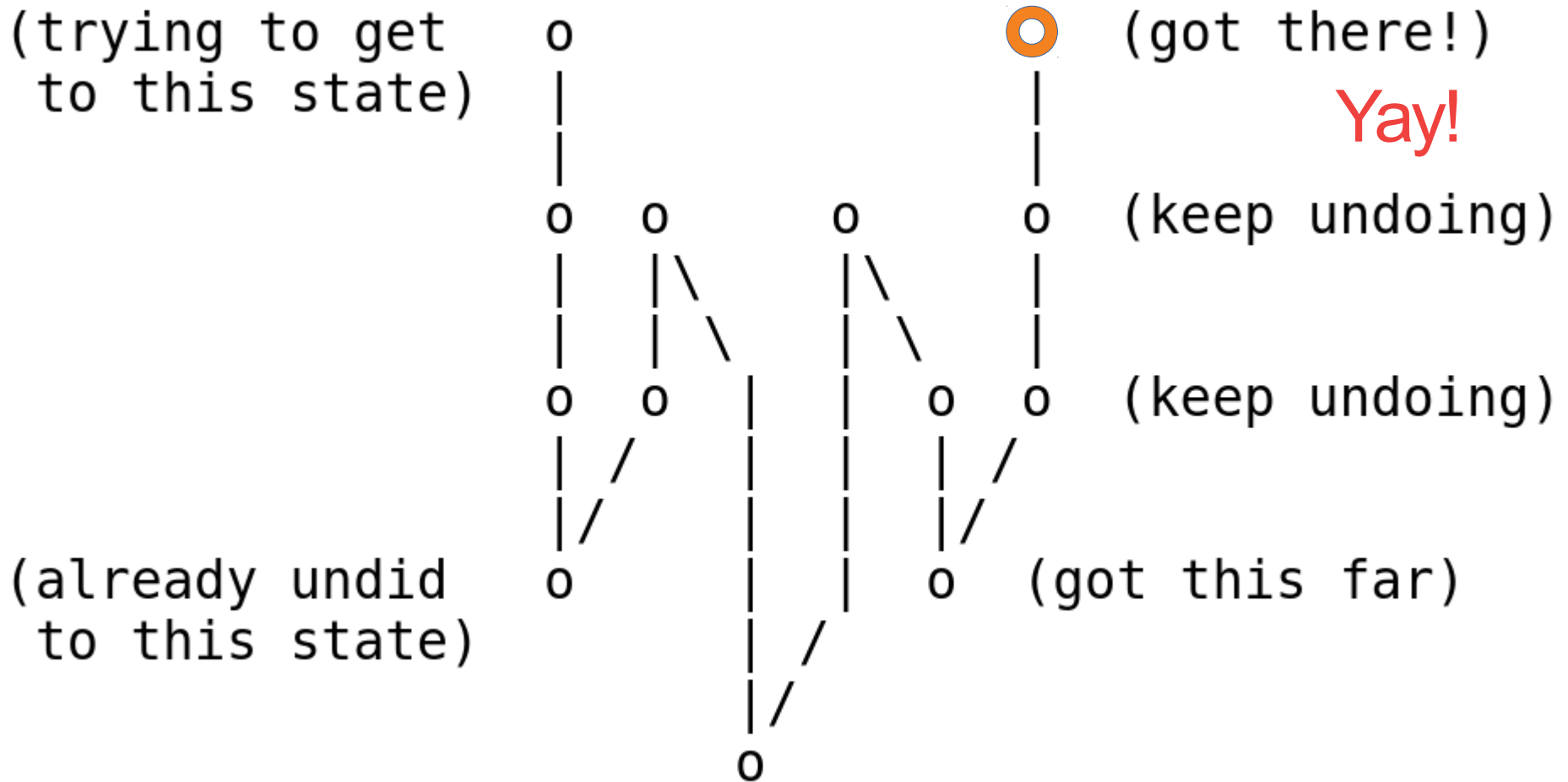
# Linear systems \_ emacs wins, but...



# Linear systems \_ emacs wins, but...

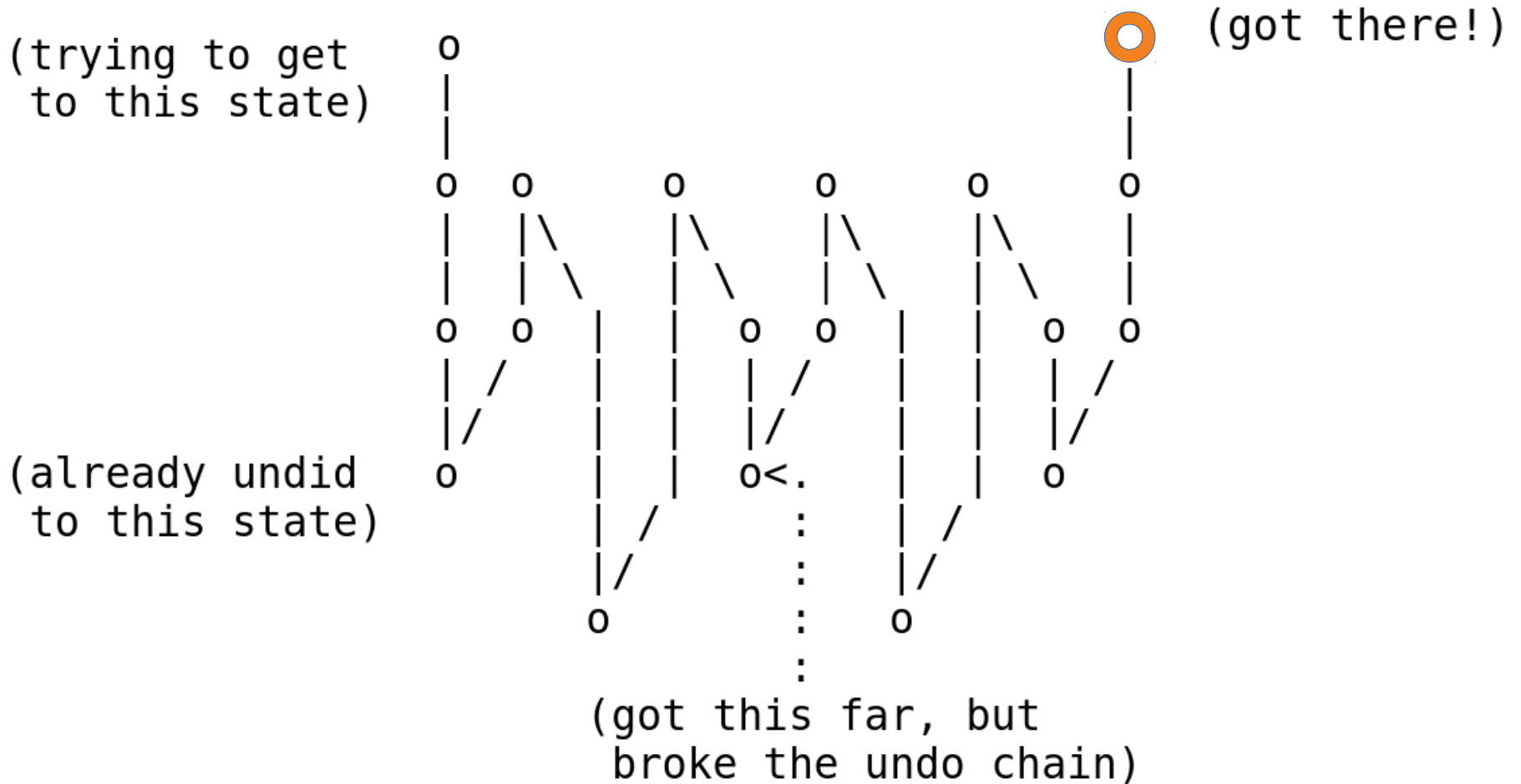


# Linear systems \_ emacs wins, but...

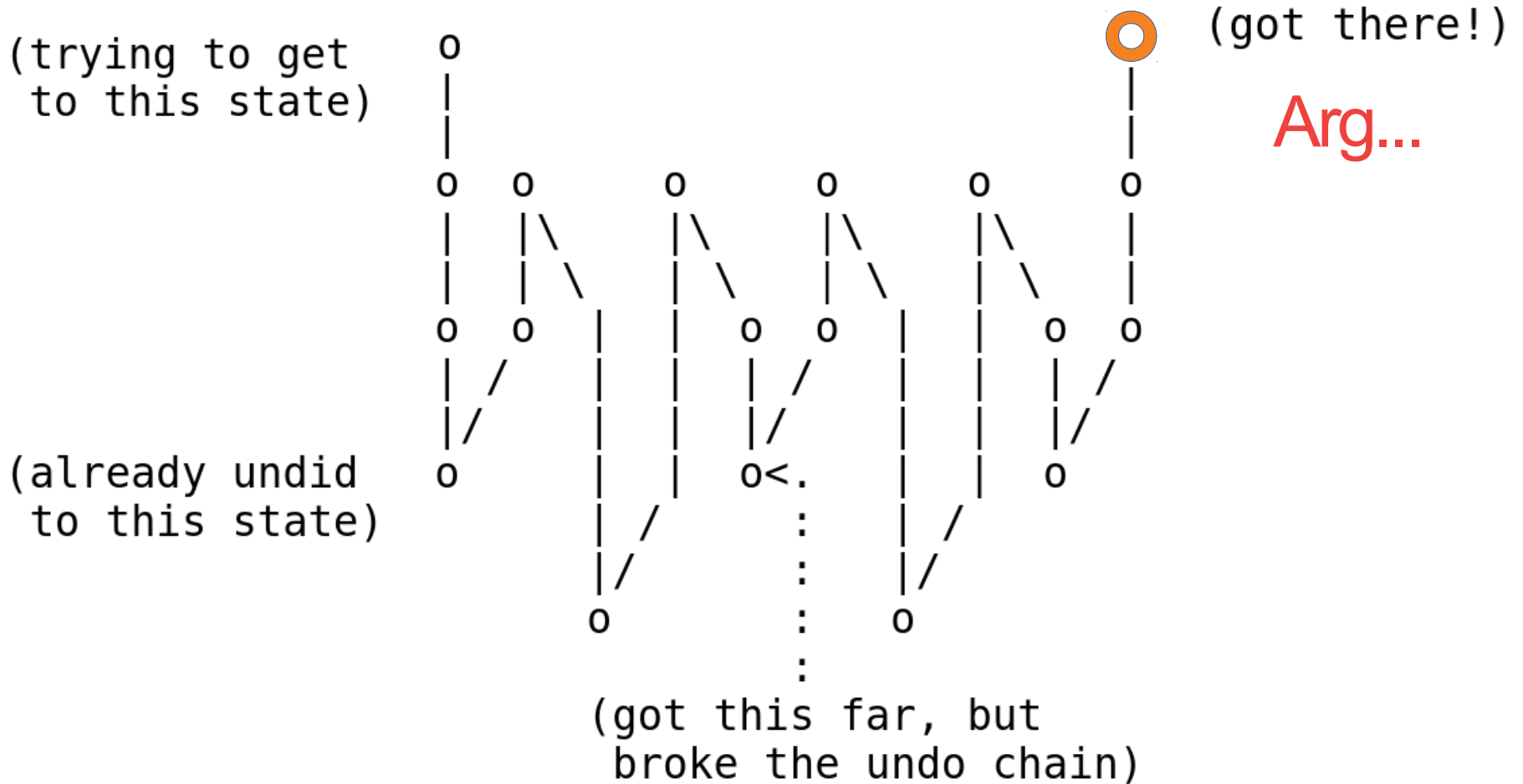




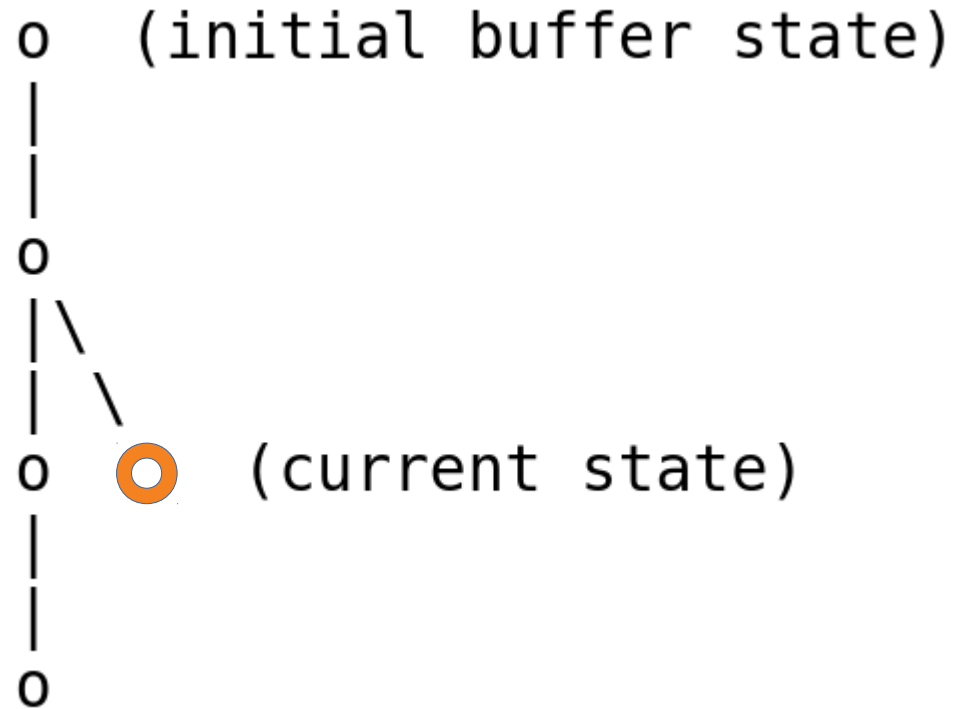
# Linear systems \_ emacs wins, but...



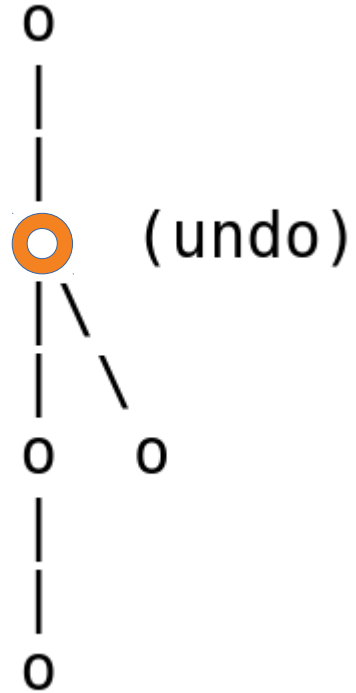
# Linear systems \_ emacs wins, but...



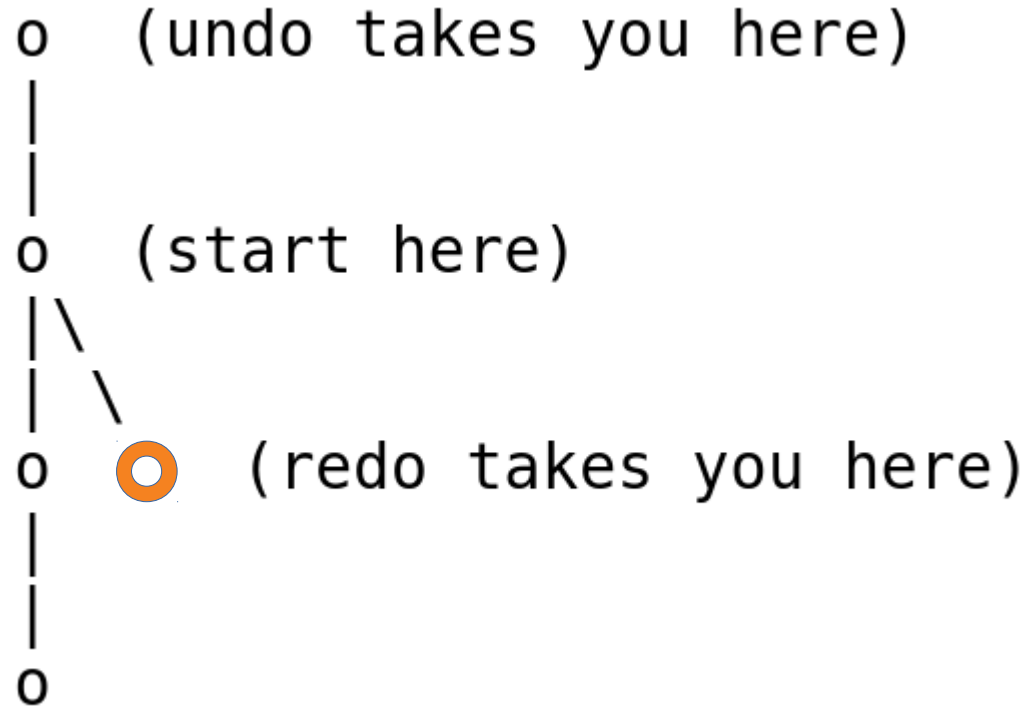
# Non linear system \_ undo-tree



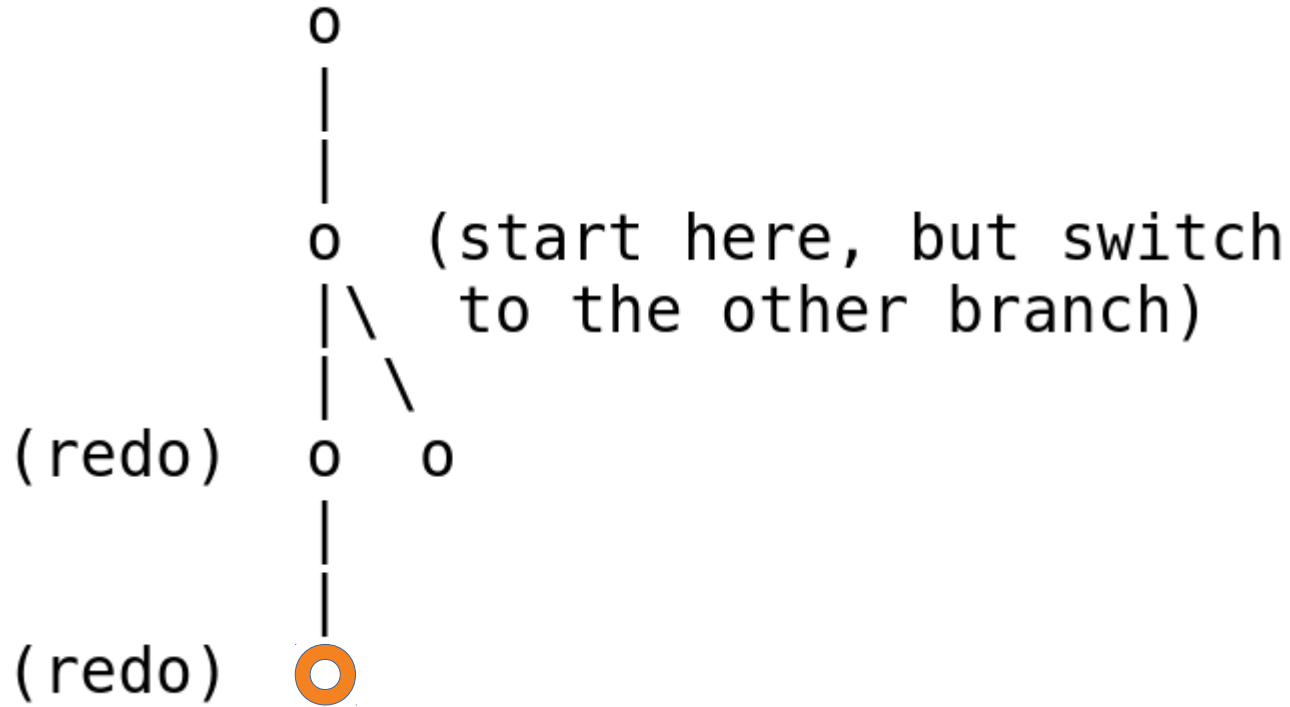
# Non linear system \_ undo-tree



# Non linear system \_ undo-tree



# Non linear system \_ undo-tree



# Non linear system \_ undo-tree

