











## Parsing for the Turing Test

- Most linguistic properties are defined over trees.
- One needs to parse to see subtle distinctions. E.g.:

Sara dislikes criticism of her. $(her \neq Sara)$ Sara dislikes criticism of her by anyone. $(her \neq Sara)$ Sara dislikes anyone's criticism of her. $(her = Sara \text{ or } her \neq Sara)$ 

 In rest of lecture (and following two lectures), we'll develop some parsing algorithms on the blackboard.

## CKY algorithm, recognizer version

- Input: string of n words
- Output: yes/no (since it's only a recognizer)
- Data structure: n x n table
  - rows labeled 0 to n-1
  - columns labeled 1 to n
  - cell [i,j] lists constituents found between i and j

## CKY algorithm, recognizer version

- for i := 1 to n
- Add to [i-1,i] all categories for the i<sup>th</sup> word
- for width := 2 to n
  - for start := 0 to n-width
    - Define end := start + width
    - for mid := start+1 to end-1
      for every constituent X in [start,mid]
      - for every constituent X in [start, ind]
        for every constituent Y in [mid,end]
      - for all ways of combining X and Y (if any)
      - Add the resulting constituent to
        - [start,end] if it's not already there.