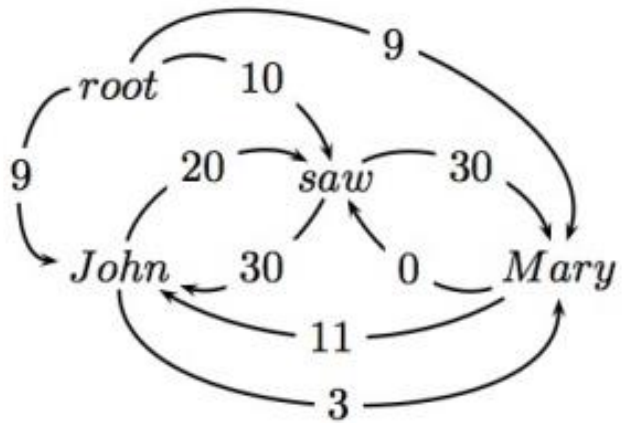
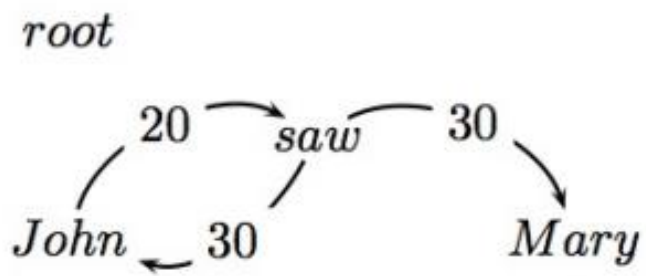
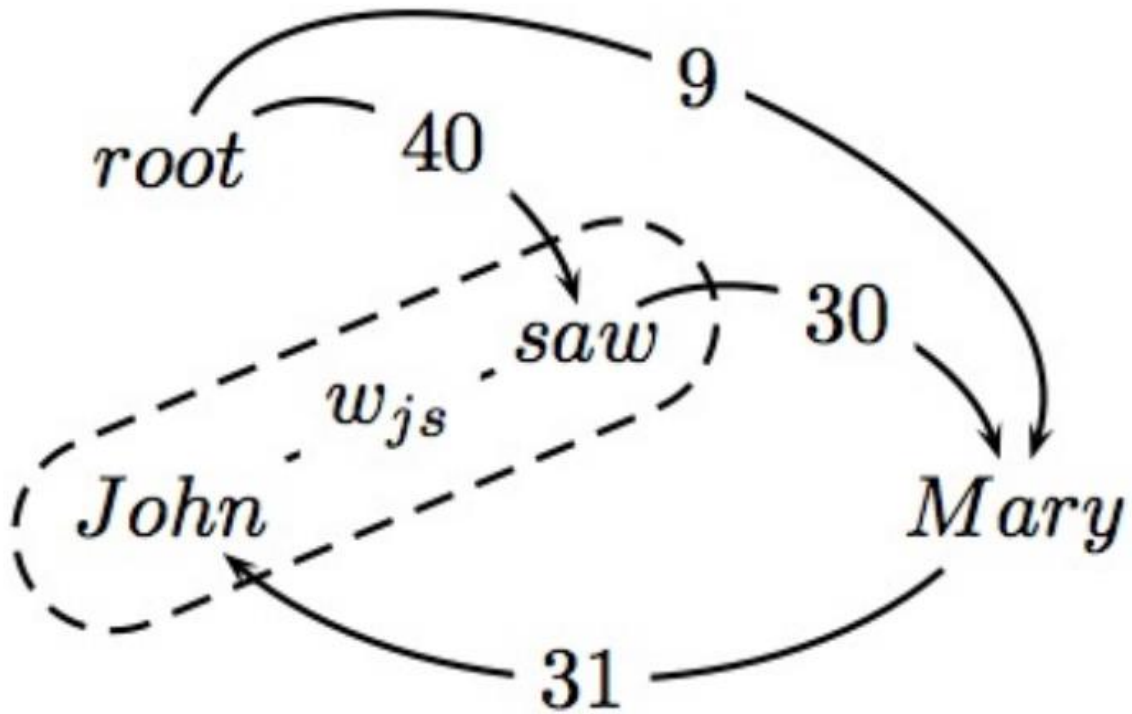
Chu-Liu-Edmonds



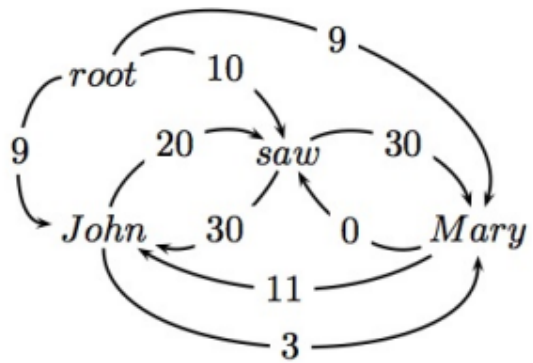


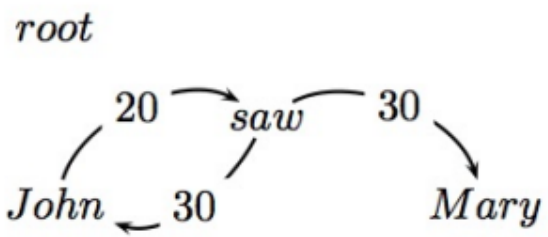
Chu-Liu-Edmonds

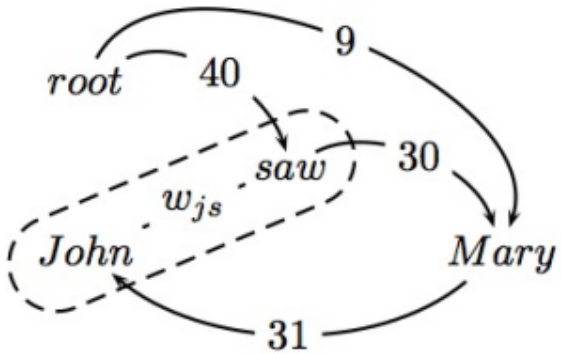
- If not a tree, identify cycle and contract
- Recalculate arc weights into and out-of cycle



Chu-Liu-Edmonds

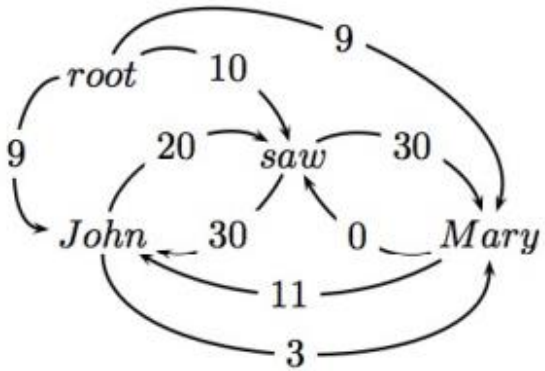


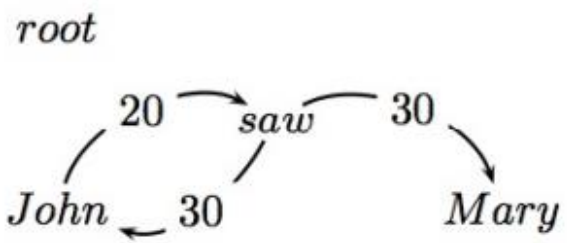


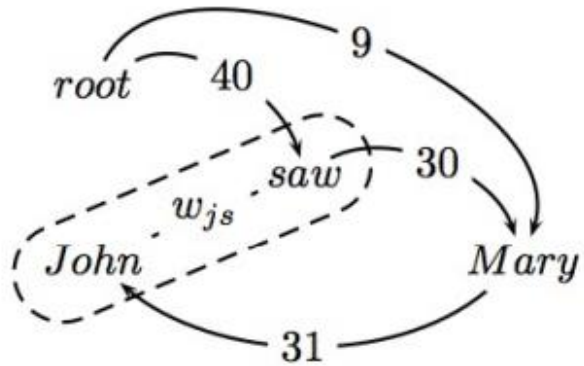


- Outgoing arc weights
  - Equal to the max of outgoing arc over all vertexes in cycle
  - e.g., John $\rightarrow$ Mary is 3 and saw $\rightarrow$ Mary is 30

Chu-Liu-Edmonds





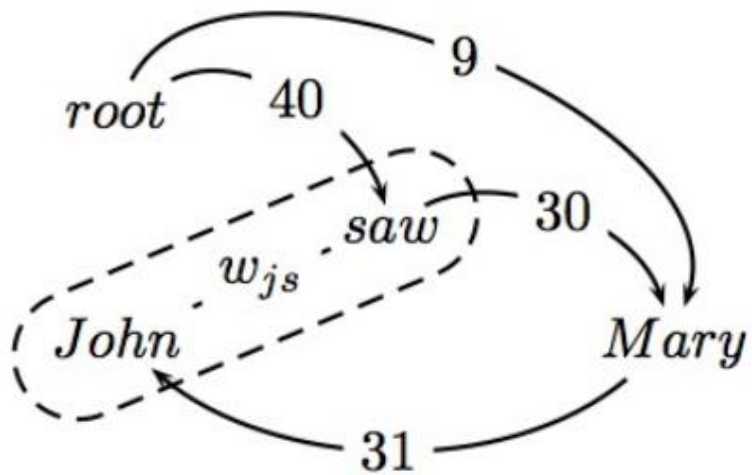


- Incoming arc weights
  - Equal to the weight of best spanning tree that includes head of incoming arc, and all nodes in cycle
  - root $\rightarrow$ saw $\rightarrow$ John is 40 (**)
  - root $\rightarrow$ John $\rightarrow$ saw is 29

Chu-Liu-Edmonds

Theorem

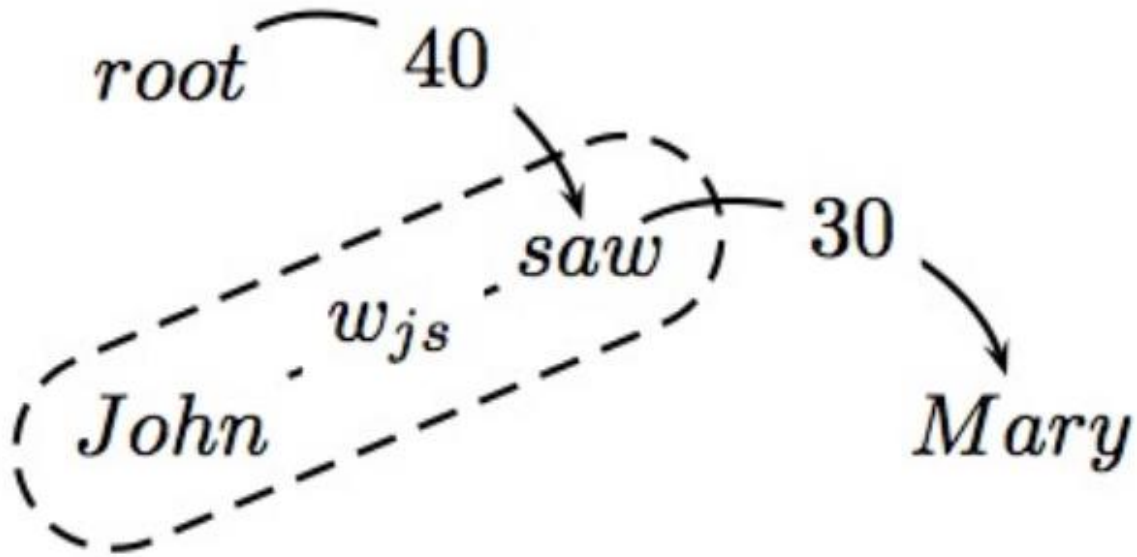
The weight of the MST of this contracted graph is equal to the weight of the MST for the original graph



- Therefore, recursively call algorithm on new graph

Chu-Liu-Edmonds

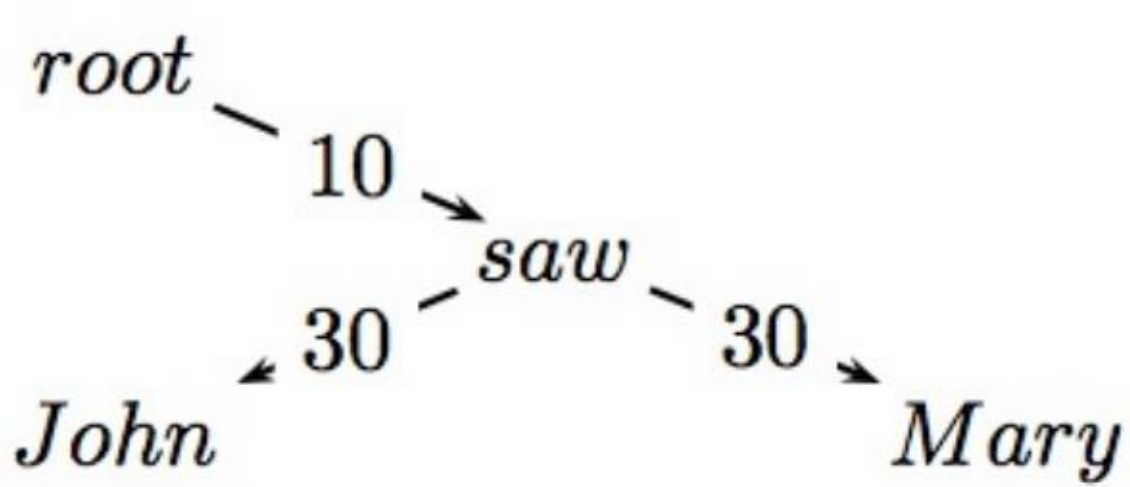
- This is a tree and the MST for the contracted graph!!



- Go back up recursive call and reconstruct final graph

Chu-Liu-Edmonds

- This is the MST!!



Learning weights of the graph

- Use training dataset
- Define features

- Step1: Construct a graph with all arcs
- Step2: initialize a weight vector based on number of features, w(0)
- Step3: find feature vector for every arc
- Step 4: find weight of every arc w_{ij} (k), by multiplying feature vector of the arc and the weight vector w(i)
- Step5: Apply CLE algo. To find MST G'
- Step6: update the w(i+1)=w(i)+f(G_t)-f(G')
- Step7: Iterate

Suppose you are training MST Parser for dependency and the sentence, “John saw Mary” occurs in the training set. Also, for simplicity, assume that there is only one dependency relation, “rel”. Thus, for every arc from word w_i to w_j, your features may be simplified to depend only on words w_i and w_j and not on the relation label. Below is the set of features

f 1 : pos(w_i) = Noun and pos(w_j) = Noun

f 2 : pos(w_i) = Verb and pos(w_j) = Noun

f 3 : w_i = Root and pos(w_j) = Verb

f 4 : w_i = Root and pos(w_j) = Noun

f 5 : w_i = Root and w_j occurs at the end of sentence

f 6 : w_i occurs before w_j in the sentence

f 7 : pos(w_i) = Noun and pos(w_j) = Verb

The feature weights before the start of the iteration are: {3, 20, 15, 12, 1, 10, 20}.

Determine the weights after an iteration over this example.

a) [3, 19, 11, 15, 3, 12, 19]
b) [3, 20, 13, 10, 3, 11, 21]
c) [4, 20, 15, 10, 1, 7, 21]
d) [3, 19, 16, 11, 1, 9, 19]