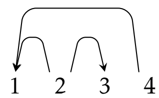
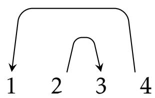
1 Connectedness

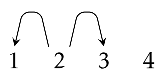
An example of a (weakly) *connected* (unlabeled dependency) graph is $G = (V, A) = (\{1,2,3,4\}, \{(2,1),(2,3),(4,1)\})$:

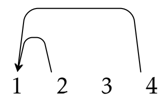


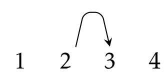
This is becuase for every $i \in V$ and $j \in V$ there is an undirected path from i to j in G. For instance, one can get from node 3 to node 1 following the arcs $(2,3)$ and $(2,1)$. Note that, even though the arcs are directed, their direction are disregarded in the definition of connectedness.

Removing even one of the arcs from A makes it disconnected. For instance, all the following graphs are disconnected:





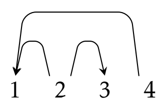




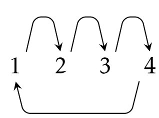
In the first graph on the left, there's no path between 1 and 2 (in particular). In the second graph, there are two connected components, $\{1,2,3\}$ and $\{4\}$ respectively, but there's no path between 4 on the one hand and 1, 2, or 3 on the other hand.

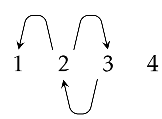
2 Acyclicity

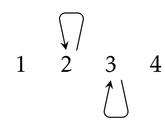
The first graph mentioned above is not only connected by also *acyclic*, since there is no cycle in this graph:



All the following are cyclic:

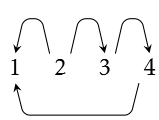






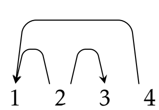
In the first one, the cycle involves all the arcs $(1,2)$, $(2,3)$, $(3,4)$, and $(4,1)$. In the second one, there is a cycle composed from two arcs, $(2,3)$ and $(3,2)$, which allows to reach. In the third graph, there are two (trivial) cycles, $(2,2)$ and $(3,3)$.

Note that the direction of the arcs matters, hence the following graph is acyclic:

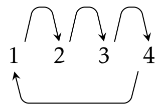


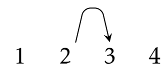
3 Single-head

In our leading example there are two arcs, $(2,1)$ and $(4,1)$, whose second component is 1. Thus node 1 has two heads specified in this graph and, therefore, the graph does not safisty the *single-head* property:



Here are two examples of graphs which, even though they are not acyclic and not connected, respectively, satisfy the single-head property.





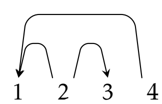
4 Projectivity

To paraphrase the definition from the lecture, a dependency graph is *projective* if it satisfies the following condition for every $i, i', j \in V$:

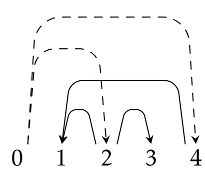
$$i \to j \land (i < i' < j \lor j < i' < i) \quad \Longrightarrow \quad i \to^* i' \tag{1}$$

In words, if there is an arc from i in j, then any node i' between i and j must be also in the projection[1] of i, i.e., there must be a directed path from i to i'.

There is a close correspondence between projectivity and a related property: *planarity*. Informally, a dependency graph is planar if it is possible to draw it without the arcs crossing (while drawing all the arcs above the nodes). However, one must be careful with applying this correspondence to determine whether a graph is projective. For instance, our leading example:



might seem projective at first sight (since clearly there are no crossing arcs). Nevertheless, it is not projective, since there is an arc from 4 to 1 and yet 2 (which is between 1 and 4 after all) is not in the projection of 4 ($\neg(4 \to^* 2)$). Hence, to benefit from the correspondence between projectivity and planarity,[2] one should first add the dummy root 0 to the graph and an arc from 0 to every i such that i has no specified head in the graph yet. This way our example becomes:



[1] A projection of a node i is the set of nodes reachable from i via a directed path: $\{j \in V : i \to^* j\}$.

[2] Another property that should be satisfied is acyclycity: if a dependency graph is not acyclic, then the equivalence between projectivity and planarity does not hold.