Exercise 1

Consider the following XML document:

```xml
<?xml version="1.0" encoding="iso-8859-1" standalone="yes"?>
1  <Auftraege>
2    <Auftrag>
3      <Kunde>Meier</Kunde>
4      <PC>pc500</PC>
5    </Auftrag>
6    <Auftrag>
7      <Kunde>Reich</Kunde>
8      <PC>pc600</PC>
9    </Auftrag>
10  </Auftraege>
```

Draw the document tree of the above given document. (The line numbers can be ignored)

Exercise 2 (Tree Pattern)

Draw the tree pattern for the following XPath queries:

2.1 /Auftraege/Auftrag/PC

2.2 //Kunde/*

2.3 /Auftraege/Auftrag[Kunde='Meier']

2.4 //Auftrag[Kunde and PC]/PC

2.5 //*/Auftrag[Kunde='Meier']/Kunde

2.6 //Auftrag[Kunde='Reich'][PC != 'pc600']

2.7 //Auftrag[PC='pc600']/*
**Exercise 3 (Embedding)**

Consider the following definition of an embedding of a tree pattern into a document tree:

Let TQ be the tree pattern of the user query Q. Let TD be the tree representation of the XML document. An embedding from TQ to TD is defined to be a function e:Nodes(TQ) \rightarrow Nodes(TD) with

1. \(e(\text{root}(TQ)) = \text{root}(TD)\)
2. \(\forall x \in \text{Nodes}(TQ): \text{Label}(x) = '*' \text{ or Label}(x) = \text{Label}(e(x))\)
3. \(\forall x,y \in \text{Nodes}(TQ): (a) \text{ if } (x,y) \text{ is a child edge in } TQ \text{ then } (e(x), e(y)) \text{ is a child edge in } TD \text{ and } (b) \text{ if } (x,y) \text{ is a descendant edge in } TQ \text{ then } e(y) \text{ is a descendant of } e(x) \text{ in } TD.\)
4. \(\forall x \in \text{Nodes}(TQ), \forall y \in \text{ComparisonNodes}(TQ): \text{ if } (x,y) \text{ is an edge in } TQ, \text{ then } e(x) \text{ has to fulfill the condition stated in } \text{label}(y).\)

Calculate the set of embeddings of each tree pattern drawn in Exercise 2. First, number the nodes of the tree pattern in the order in which they occur within the query string, and number the nodes of the XML tree in pre-order. Then you can state one possible embedding of the tree pattern of query 2.1 for example by:

\[e = \{(1,1), (2,2), (3,4)\}\]

which means, that node number 1 of the tree pattern is embedded into node number 1 of the document tree, node number 2 of the tree pattern is embedded into node number 2 of the document tree, and node number 3 of the tree pattern is embedded into node number 4 of the document tree.

**Exercise 4 (Homomorphism)**

Consider following definition of homomorphism between two tree pattern:

Let TQ and TA be tree patterns The homomorphism is a function h:Nodes(TA) \rightarrow Nodes(TQ) so that

1. \(h(\text{root}(TA)) = \text{root}(TQ)\)
2. \(\forall x \in \text{Nodes}(TA): \text{Label}(x) = '*' \text{ or Label}(x) = \text{Label}(h(x))\)
3. \(\forall x,y \in \text{Nodes}(TA): (a) \text{ if } (x,y) \text{ is a child edge in } TA \text{ then } (h(x), h(y)) \text{ is a child edge in } TQ \text{ and } (b) \text{ if } (x,y) \text{ is a descendant edge in } TA \text{ then there exists a path from } h(x) \text{ to } h(y) \text{ in } TQ.\)

Draw the tree pattern of the following queries:

4.1 a//*/b
4.2 a//b
4.3 a/*/b
4.4 a/x/b

and check for homomorphisms between them.
Exercise 5 (XPath Looking Forward)

(a) Warm up: Rewrite the following XPath expressions, such that they do not contain any backward-axes (i.e., no parent, ancestor, ancestor-or-self, preceding, preceding-sibling axes). Try to avoid joins wherever possible

5.1 /descendant::price/preceding::name
5.2 /child::journal/child::editor/parent::*
5.3 /descendant::name[ancestor::journal]
5.4 /descendant::name[preceding-sibling::name]/parent::authors/ancestor::journal/child::title

(b) Advanced: Salic Law

In the old times, the decision on the succession to the throne of a kingdom, in case the old king passes away, was sometimes defined by the so-called Salic Law. This law stipulates that the new king is the first living man descending via exclusively a male line from the old king such that (1) all its male ancestors have passed away, and (2) it has no older brother that lives or has a living male descendant via exclusively a male line.

Consider the genealogical tree of the descendants of the old king as an XML document with the following DTD:

<!ELEMENT oldking (name, (man | woman)* alive?)>
<!ELEMENT man (name, (man | woman)* alive?)>
<!ELEMENT woman (name, (man | woman)* alive?)>
<!ELEMENT name (#PCDATA)>

One possible formalization of the Salic Law would be the following XPath expression:

/descendant::man[child::alive]
[not ancestor::woman=/descendant::woman]
[not preceding-sibling::man/child::alive]
[not preceding-sibling::man/descendant::man[child::alive]
[not ancestor::woman=/descendant::woman]
]

Rewrite this XPath expression, such that it does not contain any backward-axes.