Assignment 1a skeleton

Submit your solution as *h1a.ipynb*. Also submit a run of the notebook *h1a.pdf*.

```
In [ ]: from nltk.tree import Tree
```

```
In [ ]: t1 = Tree.fromstring("(S (NP I) (VP (V saw) (NP him)))")
```

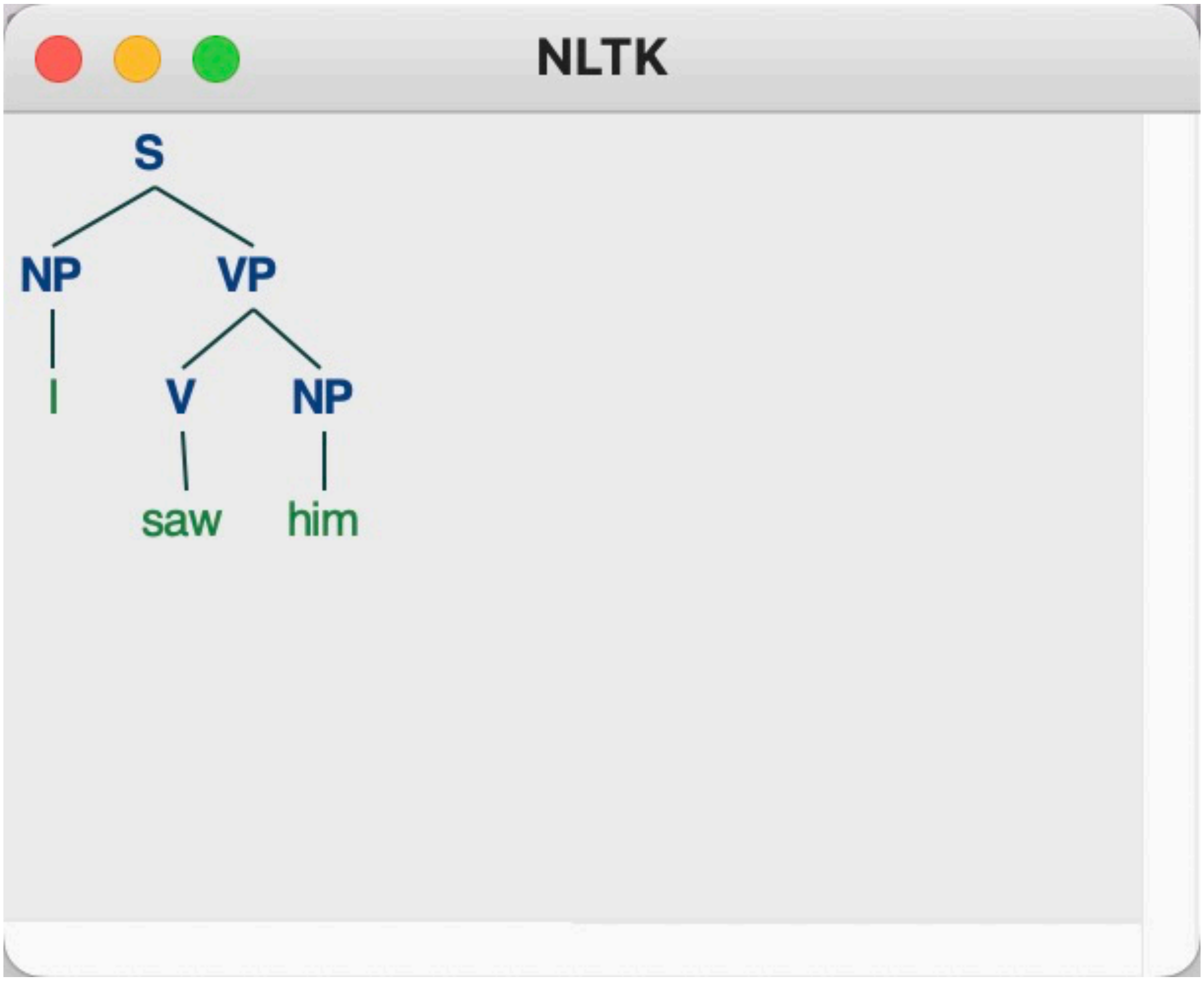
Problem 1. Display t1 inline as a labeled bracketing (see the partial solution h1a.pdf for the target).

```
In [ ]: print(t1)
```

```
(S (NP I) (VP (V saw) (NP him)))
```

Problem 2. Display t1 inline as a graphical tree.

```
In [ ]: Tree.draw(t1)
```



The label at a given address can be found with an iterative notation. The following indicates that `11` is the address of the object NP node.

```
In [ ]: t1[1][1].label()
```

```
Out[ ]: 'NP'
```

Problem 3. Write an analogous expression that returns the label 'V' of the verb pre-terminal.

```
In [ ]: t1[1][0].label()
```

```
Out[ ]: 'V'
```

Problem 4. There is a method that finds the yield (ordered list of leaves or terminals) for a given tree. Find it by saying `help(t1)`, and use the method to find the list of leaves for tree `t1`. The syntax is `t1.xyz()`, where `xyz` is the method name.

In [ ]:

```
help(t1)
```

In [ ]:

```
t1.leaves()
```

Out[ ]:

```
['I', 'saw', 'him']
```

Problem 5. There is a method that finds a python representation of the tree domain (collection of addresses) for a tree. Use it to find a representation of the tree domain for `t1`. This initial result should be a list.

In [ ]:

```
# Incorrect solution
t1.treepositions()
```

Out[ ]:

```
[(), (0,), (0, 0), (1,), (1, 0), (1, 0, 0), (1, 1), (1, 1, 0)]
```

According to Lecture 1, a tree domain is a set of addresses rather than a list of addresses. Use the python functionality for converting lists to sets to fix this.

In [ ]:

```
set(t1.treepositions())
```

Out[ ]:

```
{(), (0,), (0, 0), (1,), (1, 0), (1, 0, 0), (1, 1), (1, 1, 0)}
```

In [ ]:

```

```