Assignment 1a

This is Mats Rooth's solution. The distributed file h1a.pdf is derived from this by whiting out some parts.

In [1]:
```python
from nltk.tree import Tree
```

In [2]:
```python
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 [3]: [redacted]

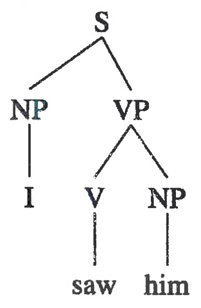
```
(S (NP I) (VP (V saw) (NP him)))
```

Problem 2. Display t1 inline as a graphical tree.

(This is a likely locus for problems related to the python and jupyter intstallation, which result in the graphics not working. See the forum for ideas about how to solve them.)

In [4]: [redacted]

Out[4]:



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 [5]:
```python
t1[1][1].label()
```

Out[5]:
```
'NP'
```

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

In [6]: [redacted]

Out[6]:
```
'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)` . Use the method to find the list of terminal words for tree `t1` . The syntax is `t1.xyz()` , where `xyz` is the method name.

```
In [7]: [redacted]
```

```
Out[7]: ['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 [8]: [redacted]
```

```
Out[8]: [redacted]
```

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 [9]: [redacted]
```

```
Out[9]: {(), (0,), (0, 0), (1,), (1, 0), (1, 0, 0), (1, 1), (1, 1, 0)}
```

```
In [ ]:
```

```
In [ ]:
```