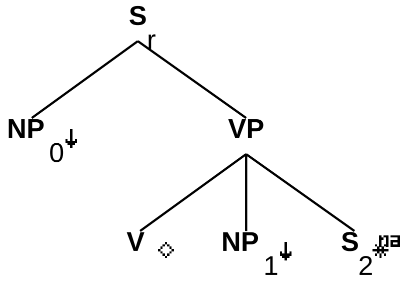
Family ”Tnx0Vnx1s2”

March 5, 2008

1 Tree ”betanx0Vnx1s2”

1.1 graphe



1.2 comments

```
Declarative tree for verbs taking an NP complement and a
sentential
complement.  The particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Max forced Bill to eat spinach.
John considers the elephant to be lazy.
John bet Mary (that) she could parachute.
```

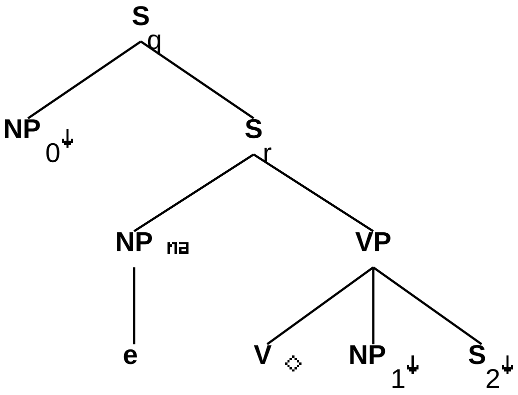
1.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
```

```
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

2 Tree ”alphaW0nx0Vnx1s2”

2.1 graphe



2.2 comments

Subject extraction tree for verbs taking an NP complement and a sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

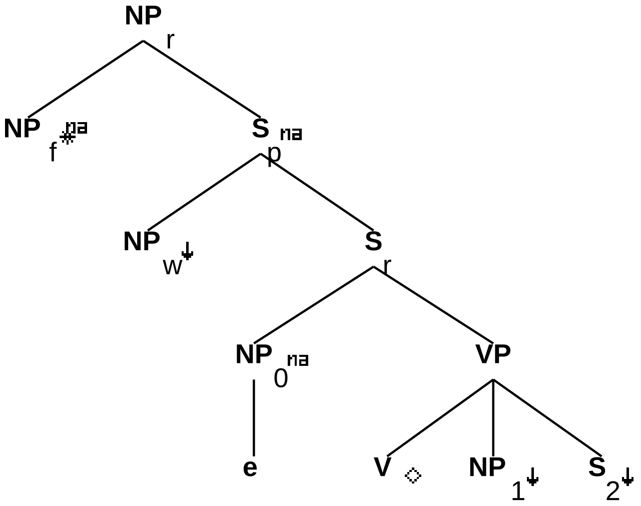
```
Exs:  Who forced Bill to eat spinach.
Which person considers the elephant to be lazy.
Which guy bet Mary (that) she could parachute.
```

2.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<wh> = NP_0:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<inv> = -
NP.t:<trace> = NP_0.t:<trace>
NP.t:<case> = NP_0.t:<case>
NP.t:<agr> = NP_0.t:<agr>
NP.t:<wh> = NP_0.t:<wh>
NP_0:<wh> = +
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-case> = NP.t:<case>
S_r.b:<agr> = NP.t:<agr>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
```

3 Tree "betaN0nx0Vnx1s2"

3.1 graphe



3.2 comments

```
Subject relative clause tree for verbs taking an NP complement and
a sentential complement.  The particular verbs place constraints on
the mode of the complement and on which complementizers may adjoin
above the complement.  Note that so-called ECM verbs get this tree
(rather than some sort of small clause analysis).

Exs:  the person who forced Bill to eat spinach
the animal that considers the elephant to be lazy
the fool who bet Mary (that) she could parachute
```

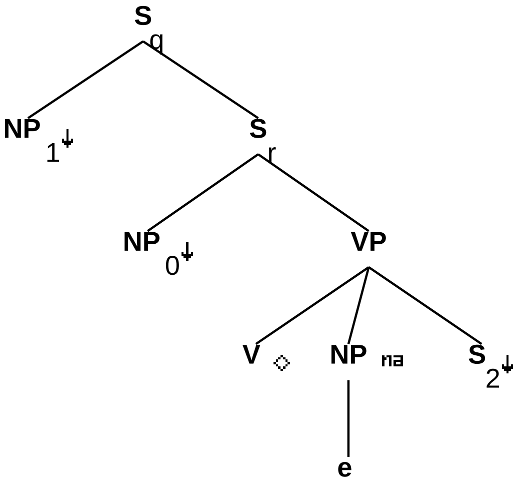
3.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>




S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
S_r.b:<agr> = NP_0:<agr>
S_r.b:<assign-case> = NP_0:<case>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
```

```
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
NP_w.t:<wh> = +
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

4 Tree "alphaW1nx0Vnx1s2"

4.1 graphe



4.2 comments

Object extraction tree for verbs taking an NP complement and a sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

Exs: Who did Max force to eat spinach?
Who did Max bet that she could parachute?

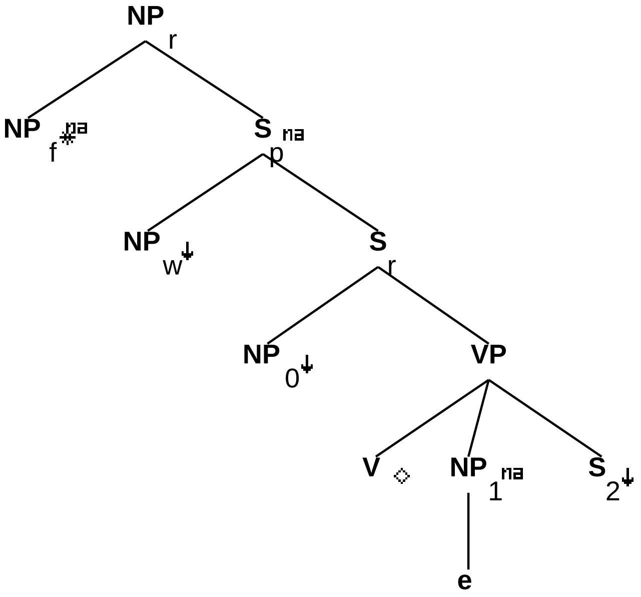
4.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<wh> = NP_1.t:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
NP_0.t:<agr> = S_r.b:<agr>
NP_0.t:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<case> = NP_1.t:<case>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<wh> = NP_1.t:<wh>
NP.t:<case> = acc
S_r.b:<tense> = VP.t:<tense>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
```

```
S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

5 Tree ”betaN1nx0Vnx1s2”

5.1 graphe



5.2 comments

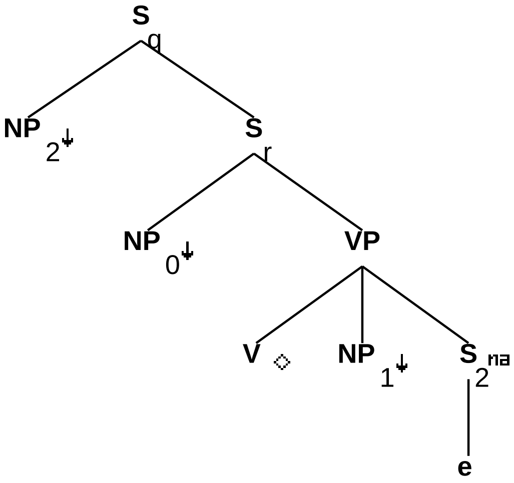
```
Object relative clause tree for verbs taking an NP complement and
a sentential complement.  The particular verbs place constraints on
the mode of the complement and on which complementizers may adjoin
above the complement.  Note that so-called ECM verbs get this tree
(rather than some sort of small clause analysis).

Exs:  the person who Max forced to eat spinach
the woman Max bet that she could parachute
```

5.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_1.t:<case> = acc
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_w.t:<wh> = +
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

6 Tree "alphaW2nx0Vnx1s2"

6.1 graphe



6.2 comments

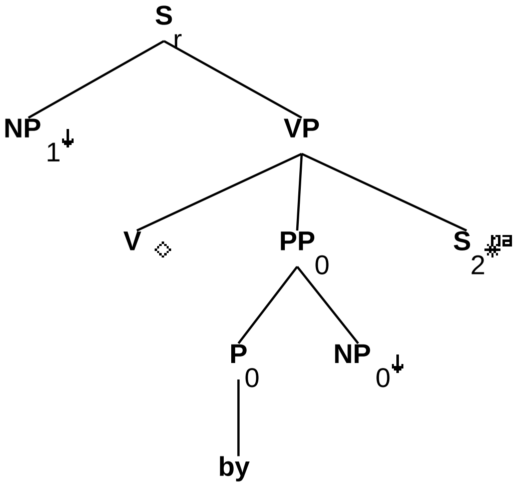
```
Sentential complement extraction tree for verbs taking an
NP complement and a sentential complement.  The particular verbs place
constraints on the mode of the complement and on which complementizers
may adjoin above the complement.  Note that so-called ECM verbs get
this tree (rather than some sort of small clause analysis).  These
sentences will also get a TW2nx0Vnx1nx2 parse (i.e. What did Max bet
Mary? A hundred dollars).  We allow both parses on the assumption that
their underlying structure is different, as are their semantic
interpretations.

Exs:  What did Max consider the elephant?
What did Max bet Mary?
```

6.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
NP_2.t:<wh> = +
NP_2.t:<case> = acc
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<wh> = NP_2:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_2:<trace> = S_2.t:<trace>
S_r.b:<tense> = VP.t:<tense>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_2.t:<assign-comp> = inf_nil/ind_nil
S_r.t:<conj> = nil
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

7 Tree ”betanx1Vbynx0s2”

7.1 graphe



7.2 comments

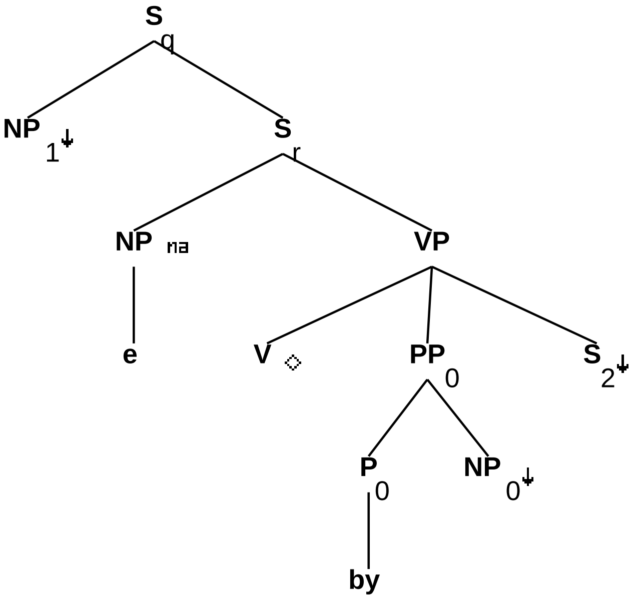
```
Passive tree for verbs taking an NP complement and a
sentential complement.  This tree has the by-phrase before the
sentential complement; there is another tree with the opposite order.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  The emu was expected by his trainer to eat oats.
Mary was bet by Bill that she could not climb Mt. Everest.
```

7.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<mode> = ind
VP.t:<passive> = +
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

8 Tree ”alphaW1nx1Vbynx0s2”

8.1 graphe



8.2 comments

Subject extraction from passive tree for verbs taking an NP complement and a sentential complement. This tree has the by-phrase before the sentential complement; there is another tree with the opposite order. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

Exs: Who was expected by his trainer to eat oats?
Who was bet by Bill that she could not climb Mt. Everest?

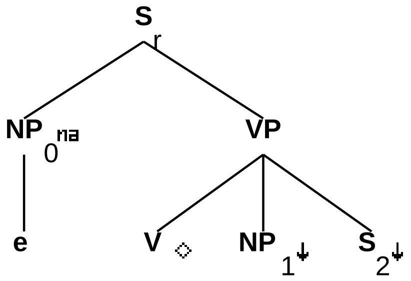
8.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<wh> = NP_1.t:<wh>
S_r.t:<comp> = nil
S_r.b:<inv> = -
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_q.b:<comp> = S_r.t:<comp>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<tense> = S_r.t:<tense>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<case> = NP_1.t:<case>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<wh> = NP_1.t:<wh>
NP_1.t:<wh> = +
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<passive> = +
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
```

9 Tree ”alphaInx0Vnx1s2”

9.1 graphe



9.2 comments

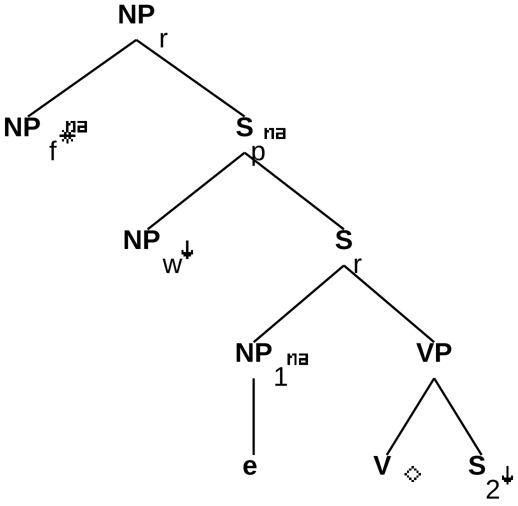
```
Imperative tree for verbs taking an NP complement and a sentential
complement.  The particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Force the crowd to stand back!
Bet them that their team will lose!
```

9.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = imp
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_0:<agr pers> = 2
NP_0:<agr 3rdsing> = -
NP_0:<agr num> = plur/sing
NP_0:<case> = nom
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<neg> = -
VP.t:<mode> = base
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.t:<tense> = pres
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

10 Tree "betaN1nx1Vs2"

10.1 graphe



10.2 comments

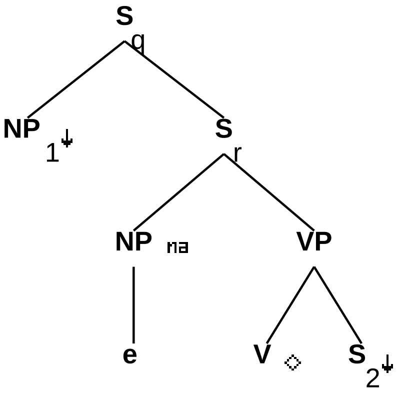
```
Subject relative clause from passive tree for verbs taking an NP
complement and a sentential complement.  This tree has no by-phrase.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  the emu that was expected to eat oats
the person who was bet that she could not climb Mt. Everest
```

10.3 features

```
NP_r.b:<wh> = NP_f.t:<wh>
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>




S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf/ppart
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<passive> = +
VP.b:<mode> = ppart
V.t:<assign-comp> = ppart_nil
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_w.t:<wh> = +
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

11 Tree "alphaW1nx1Vs2"

11.1 graphe



11.2 comments

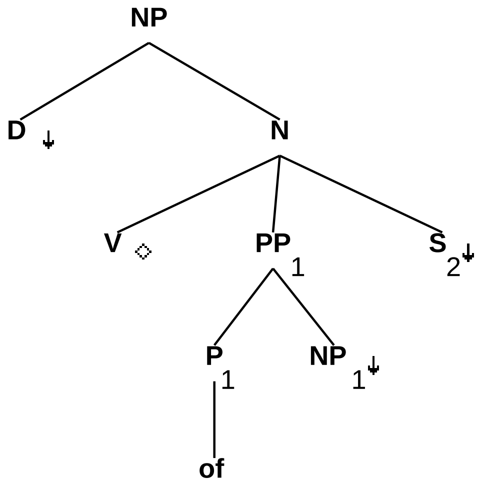
```
Subject extraction from passive tree for verbs taking an NP
complement and a sentential complement.  This tree has no by-phrase.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  which animal was expected to eat oats?
who was bet that she could not climb Mt. Everest?
```

11.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_r.t:<comp> = nil
S_r.b:<inv> = -
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<comp> = S_r.t:<comp>
S_q.b:<wh> = NP_1.t:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<wh> = NP_1.t:<wh>
NP_1.t:<wh> = +
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
```

12 Tree ”alphaDnx0Vnx1s2”

12.1 graphe



12.2 comments

Determiner Gerund tree for verbs taking an NP complement and a sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement.

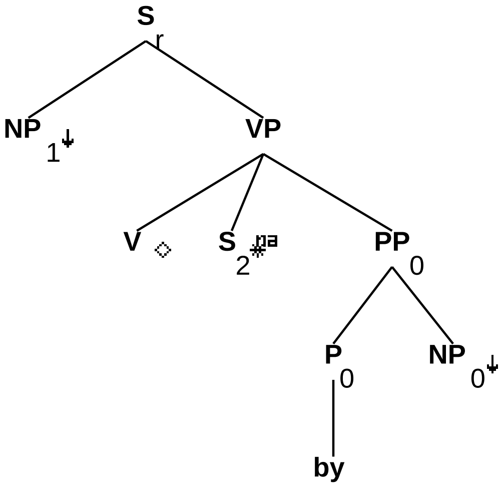
'Her forcing of the emu to dance (was cruel).'

12.3 features

```
NP.b:<const> = D.t:<const>
NP.b:<definite> = D.t:<definite>
NP.b:<quan> = D.t:<quan>
NP.b:<card> = D.t:<card>
NP.b:<gen> = D.t:<gen>
NP.b:<decreas> = D.t:<decreas>
NP.b:<wh> = D.t:<wh>
V.b:<mode> = ger
NP.b:<case> = nom/acc
NP.b:<agr num> = sing
NP.b:<agr pers> = 3
NP.b:<agr 3rdsing> = +
P_1.b:<assign-case> = acc
PP_1.b:<assign-case> = P_1.t:<assign-case>
PP_1.b:<assign-case> = NP_1.t:<case>

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
```

13 Tree "betanx1Vs2bynx0"

13.1 graphe



13.2 comments

Passive tree for verbs taking an NP complement and a sentential complement. This tree has the by-phrase at the end; there is another tree with the by-phrase before the sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

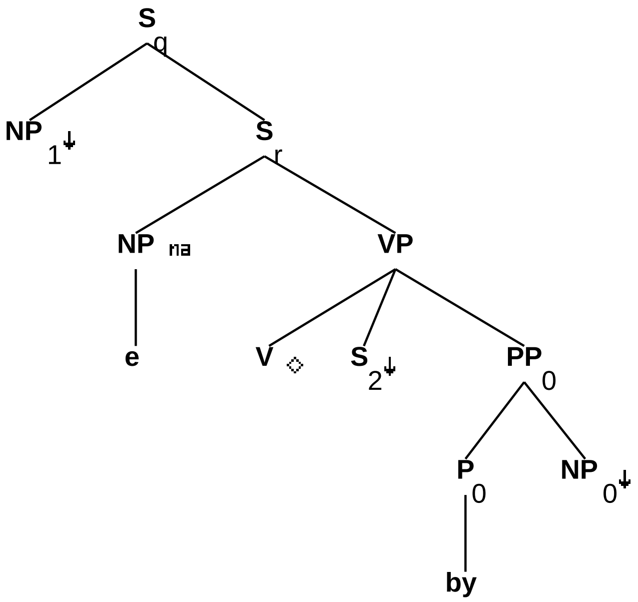
Exs: Max was forced to eat spinach by Bill.
John was considered to be lazy by the elephant.

13.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

14 Tree "alphaW1nx1Vs2bynx0"

14.1 graphe



14.2 comments

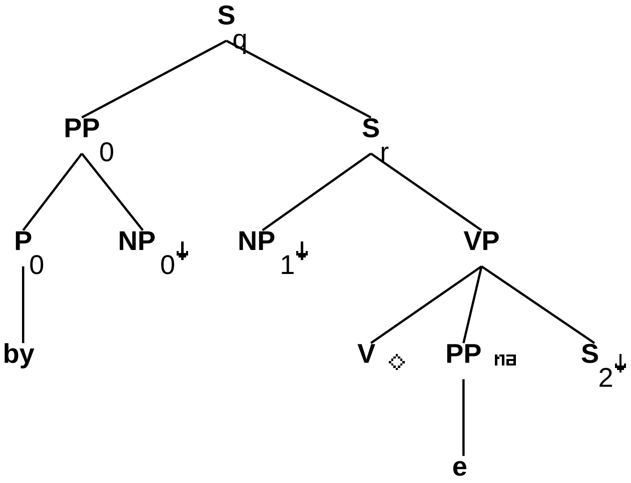
```
Passive, subject extracted tree for verbs taking an NP
complement and a sentential complement. This tree has the by-phrase at
the end; there is another tree with the by-phrase before the
sentential complement.  The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Which person was forced to eat spinach by Bill?
Who was considered to be lazy by the elephant?
```

14.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<inv> = -
S_q.b:<comp> = S_r.t:<comp>
S_q.b:<mode> = S_r.t:<mode>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
NP.t:<trace> = NP_1.t:<trace>
NP.t:<case> = NP_1.t:<case>
NP.t:<agr> = NP_1.t:<agr>
NP.t:<wh> = NP_1.t:<wh>
NP_1.t:<wh> = +
S_q.b:<wh> = NP_1.t:<wh>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = +
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
```

15 Tree "alphapW0nx1Vbynx0s2"

15.1 graphe



15.2 comments

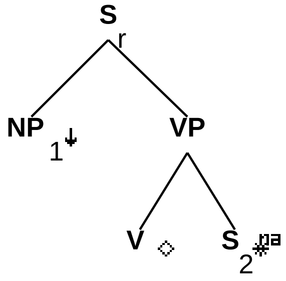
```
Passive tree for verbs taking an NP
complement and a sentential complement; extraction of by-phrase.
The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  By whom was Max forced to eat spinach?
By whom was the elephant considered to be lazy?
```

15.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<wh> = PP.t:<wh>
PP_0.b:<wh> = NP_0:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP_0:<case> = PP_0.b:<assign-case>
PP_0.t:<trace> = PP.t:<trace>

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

16 Tree "betanx1Vs2"

16.1 graphe



16.2 comments

Passive tree for verbs taking an NP complement and a sentential complement. This tree has no by-phrase. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

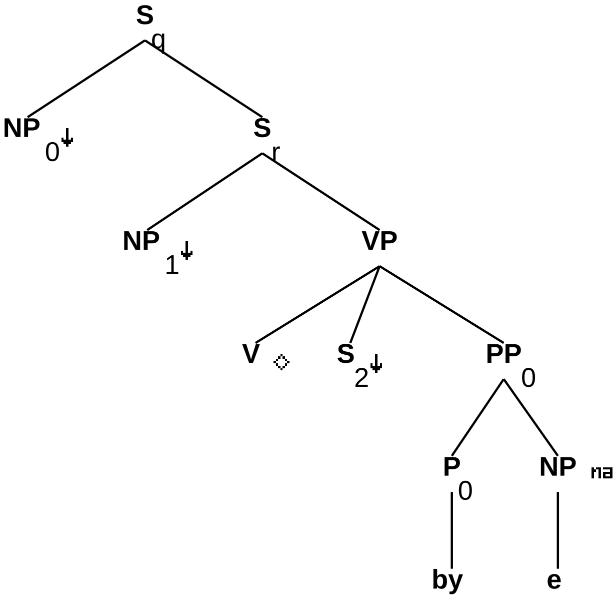
Exs: Max was forced to eat spinach.
John was considered to be lazy.

16.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

17 Tree "alphaW0nx1Vbynx0s2"

17.1 graphe



17.2 comments

```
Passive tree for verbs taking an NP
complement and a sentential complement; extraction of agent from by-phrase.
The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Who was Max forced to eat spinach by?
Which person was the elephant considered to be lazy by?

  Also, topicalization:
the coach, was John forced to run by
(I know, it sounds pretty bad, but I'm told by the
topicalization expert that it is OK)
```

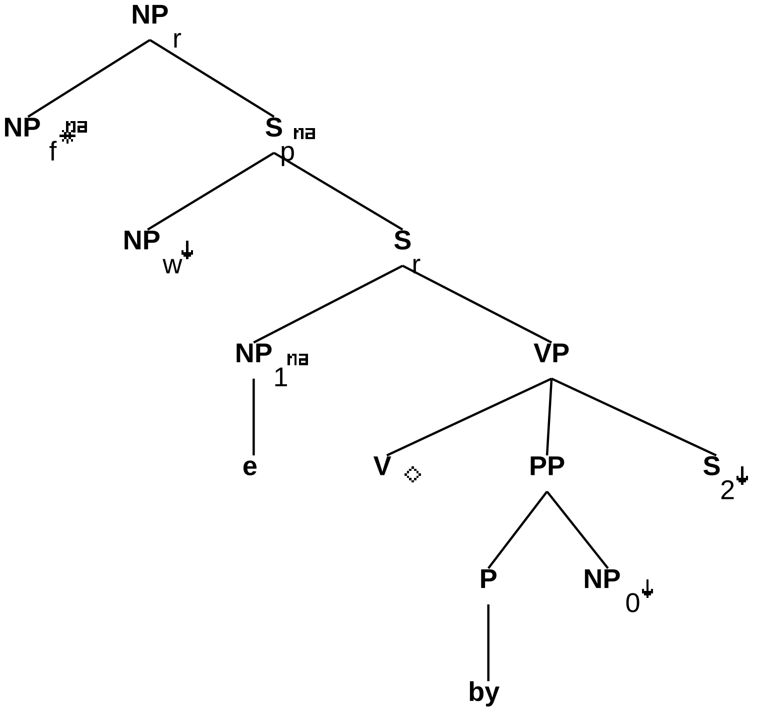
17.3 features

```
S_q.b:<extracted> = +

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_q.b:<wh> = NP_0:<wh>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.b:<inv> = -
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
NP.t:<wh> = NP_0.t:<wh>
NP.t:<agr> = NP_0.t:<agr>
NP.t:<case> = NP_0.t:<case>
NP.t:<trace> = NP_0.t:<trace>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP:<case> = PP_0.b:<assign-case>
NP:<wh> = PP_0.b:<wh>

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

18 Tree ”betaN1nx1Vbynx0s2”

18.1 graphe



18.2 comments

```
Passive tree for verbs taking an NP
complement and a sentential complement; relative clause on subject.
The by-phrase precedes the sentential complement - there is another
tree with the by-phrase after it.
The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  the person who was forced to eat spinach by Max
the animal that was considered to be lazy by the elephant
```

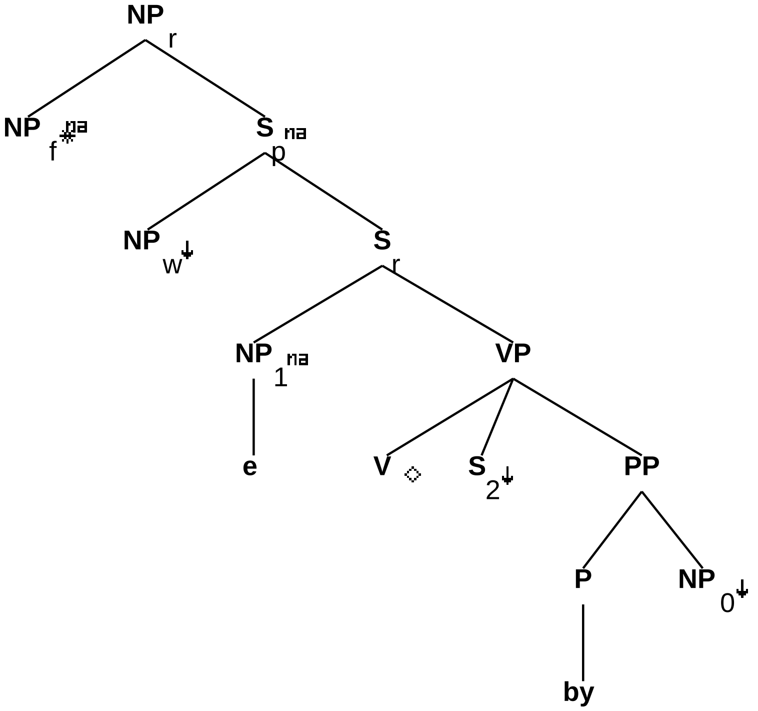
18.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind/inf/ppart
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_w.t:<wh> = +
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

19 Tree "betaN1nx1Vs2bynx0"

19.1 graphe



19.2 comments

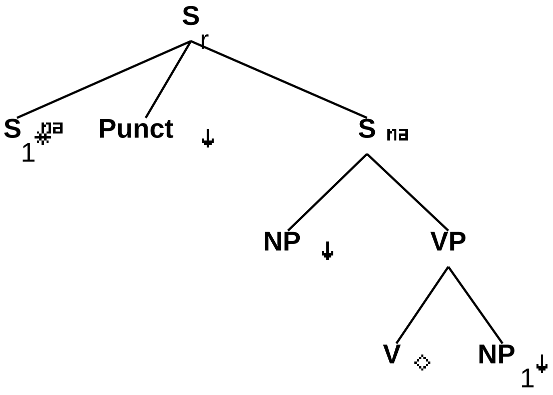
```
Passive tree for verbs taking an NP
complement and a sentential complement; relative clause on subject.
The by-phrase follows the sentential complement - there is another
tree with the by-phrase beforee it. The particular verbs place
constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  the person who was forced by Max to eat spinach
the animal that was considered by the elephant to be lazy
```

19.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>




NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.t:<mode> = ind/inf
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.t:<mode> = ind
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_w.t:<wh> = +
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

20 Tree ”betaspunxVnx1”

20.1 graphe



20.2 comments

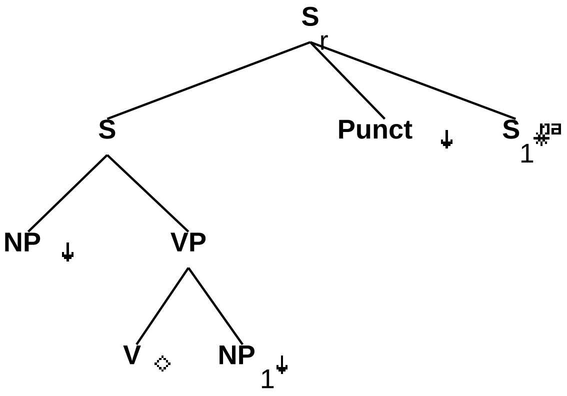
```
Post-sentential quoting clause
```

20.3 features

```
S_1.t:<comp> = nil
S_1.t:<comp> = S_r.b:<comp>
S_1.t:<extracted> = S_r.b:<extracted>
S_1.t:<assign-comp> = S_r.b:<assign-comp>
S_1.t:<tense> = S_r.b:<tense>
S_1.t:<wh> = S_r.b:<wh>
S_1.t:<inv> = S_r.b:<inv>
S_1.t:<invlink> = S_r.b:<invlink>
S_1.t:<mode> = S_r.b:<mode>
S_1.t:<assign-case> = S_r.b:<assign-case>
S_1.t:<agr> = S_r.b:<agr>


S.t:<inv> = -
S.t:<mode> = ind
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<assign-case> = VP.t:<assign-case>
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>

NP:<agr> = S.b:<agr>
NP:<case> = S.b:<assign-case>

VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense>=V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
NP_1:<case>=acc
```

21 Tree "betapunxVnx1pus"

21.1 graphe



21.2 comments

```
Verb of saying as pre-S modifier (also made NP modifier version):
  John is ill, Mary says

Also, for:

and/Conj_CONJs , they_NXN assert_punxVpuvx , any/Det_Ddx further/A_An drop/N_NXN in/Prep_n

This verb will be selected by all of the verbs of saying. It is quite
weird, though. The sentential complement of the verb (the
propositional argument) is really *distributed* in what it adjoins
onto.

Not sure about internal S features, probably should be just like rel
clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of
foot)
```

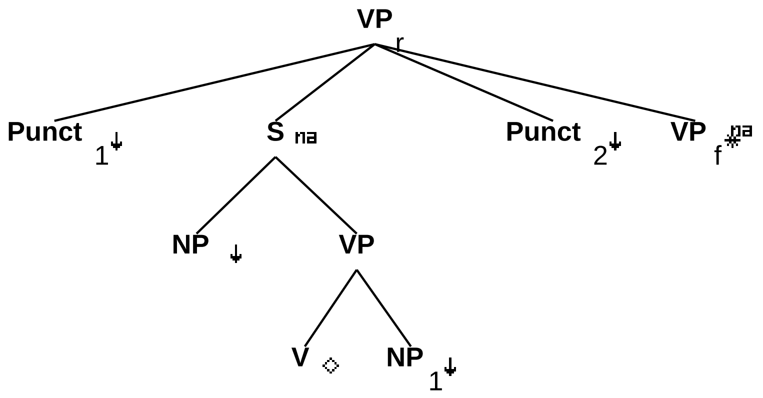
21.3 features

```
S_r.b:<punct struct> = Punct.t:<punct struct>
S_1.t:<comp> = nil
```

```
S_1.t:<comp> = S_r.b:<comp>
S_1.t:<extracted> = S_r.b:<extracted>
S_1.t:<assign-comp> = S_r.b:<assign-comp>
S_1.t:<tense> = S_r.b:<tense>
S_1.t:<wh> = S_r.b:<wh>
S_1.t:<inv> = S_r.b:<inv>
S_1.t:<invlink> = S_r.b:<invlink>
S_1.t:<mode> = S_r.b:<mode>
S_1.t:<assign-case> = S_r.b:<assign-case>
S_1.t:<agr> = S_r.b:<agr>


S.t:<inv> = -
S.t:<mode> = ind/inf
S.t:<comp> = nil
S.b:<comp> = nil
S.b:<assign-case> = VP.t:<assign-case>
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>

NP:<agr> = S.b:<agr>
NP:<case> = S.b:<assign-case>

NP:<wh> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense>=V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
NP_1:<case>=acc
```

22 Tree "betapunxVnx1puvx"

22.1 graphe



22.2 comments

```
Verb of saying, inverted, as VP modifier (also made NP modifier version):
  John, says Mary, is ill

This verb will be selected by all of the verbs of saying. It is quite
weird, though. The sentential complement of the verb (the
propositional argument) is really *distributed* in what it adjoins
onto.

Not sure about internal S features, probably should be just like rel
clause (currently is)
Has same features on VPs as ARBvx (no clue why feats on bottom of
foot)
```

22.3 features

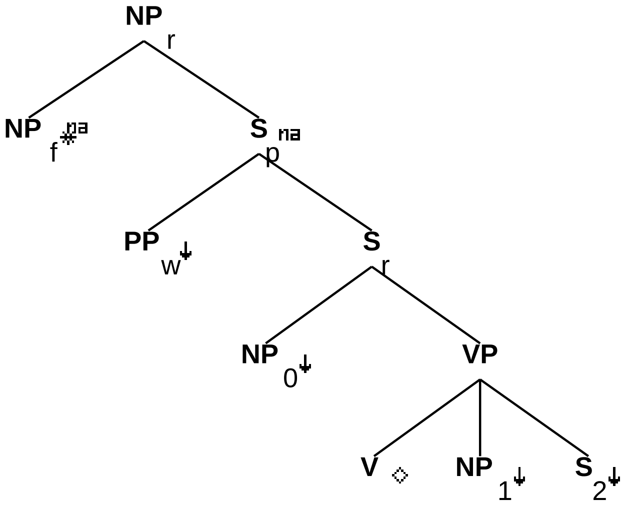
```
VP_r.b:<tense> = VP_f.t:<tense>
VP_r.b:<mode> = VP_f.t:<mode>
VP_r.b:<agr> = VP_f.t:<agr>
VP_r.b:<assign-case> = VP_f.t:<assign-case>

VP_r.b:<assign-comp> = VP_f.t:<assign-comp>
Punct_1.t:<punct struct> = Punct_2.t:<punct struct>
Punct_1.t:<punct struct> = VP_r.b:<punct struct>

S.b:<agr> = NP.t:<agr>
S.b:<assign-case> = NP.t:<case>
NP.t:<wh> = -


S.t:<mode> = ind
S.t:<comp> = nil
```

```
S.b:<comp> = nil
S.b:<agr> = VP.t:<agr>
S.b:<tense> = VP.t:<tense>
S.b:<mode> = VP.t:<mode>
S.b:<assign-case> = VP:<assign-case>
```

```
VP.b:<agr> = V.t:<agr>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<tense>=V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -

V.t:<passive> = -
V:<assign-comp> = V.t:<assign-comp>
V:<mode> = V.t:<mode>
V:<tense>=V.t:<tense>
V:<passive> = V.t:<passive>
NP_1:<case>=acc
```

23 Tree "betaNpxnx0Vnx1s2"

23.1 graphe



23.2 comments

```
Declarative tree for verbs taking an NP complement and a
sentential
```

complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

Exs: Max forced Bill to eat spinach.
John considers the elephant to be lazy.
John bet Mary (that) she could parachute.

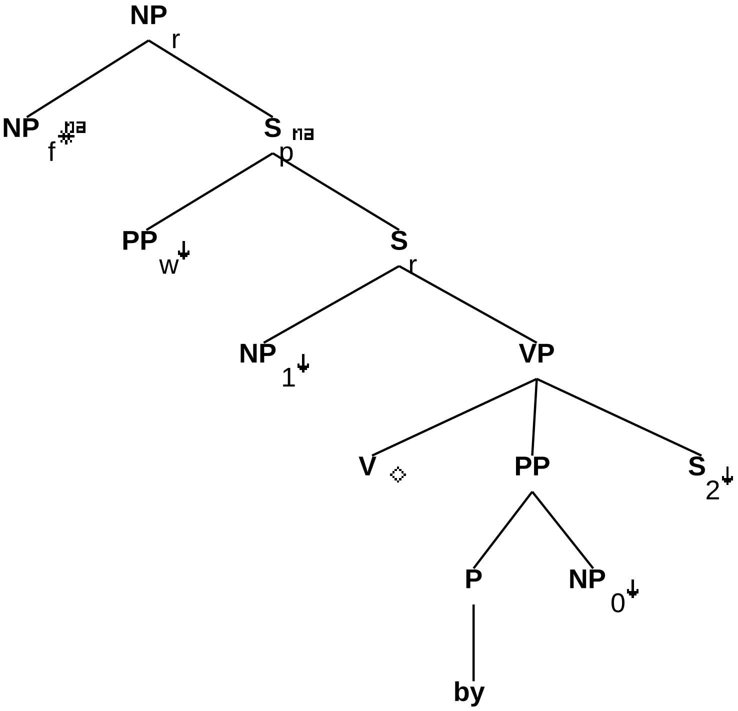
23.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
S_r.t:<inv> = -
PP_w.t:<wh> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

```
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

24 Tree ”betaNpxnx1Vbynx0s2”

24.1 graphe



24.2 comments

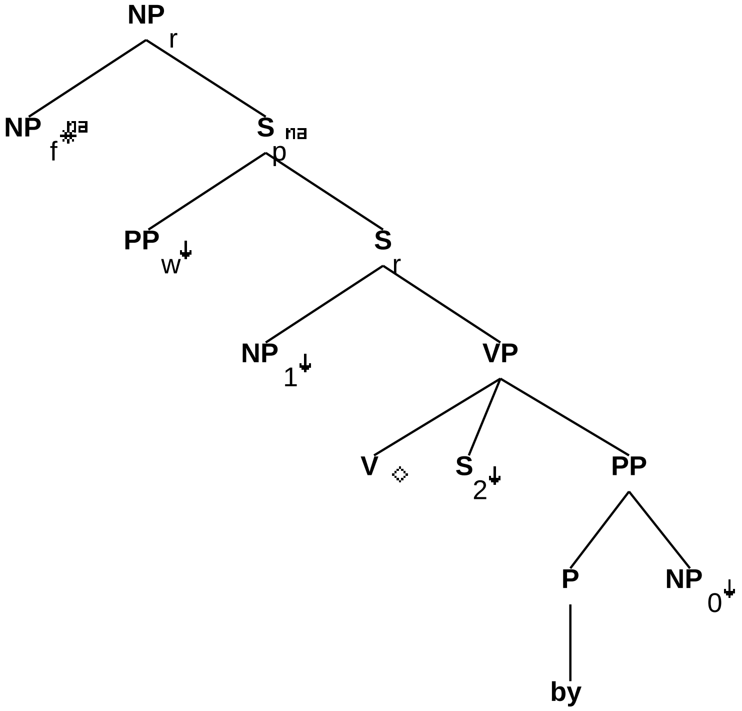
```
Passive tree for verbs taking an NP complement and a
sentential complement.  This tree has the by-phrase before the
sentential complement; there is another tree with the opposite order.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  The emu was expected by his trainer to eat oats.
Mary was bet by Bill that she could not climb Mt. Everest.
```

24.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<mode> = ind
VP.t:<passive> = +
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.t:<inv> = -
PP_w.t:<wh> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

25 Tree ”betaNpxnx1Vs2bynx0”

25.1 graphe



25.2 comments

Passive tree for verbs taking an NP complement and a sentential complement. This tree has the by-phrase at the end; there is another tree with the by-phrase before the sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

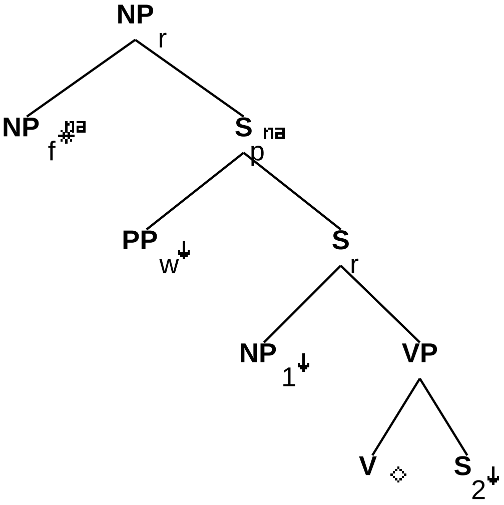
Exs: Max was forced to eat spinach by Bill.
John was considered to be lazy by the elephant.

25.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.t:<inv> = -
PP_w.t:<wh> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

26 Tree "betaNpxnx1Vs2"

26.1 graphe



26.2 comments

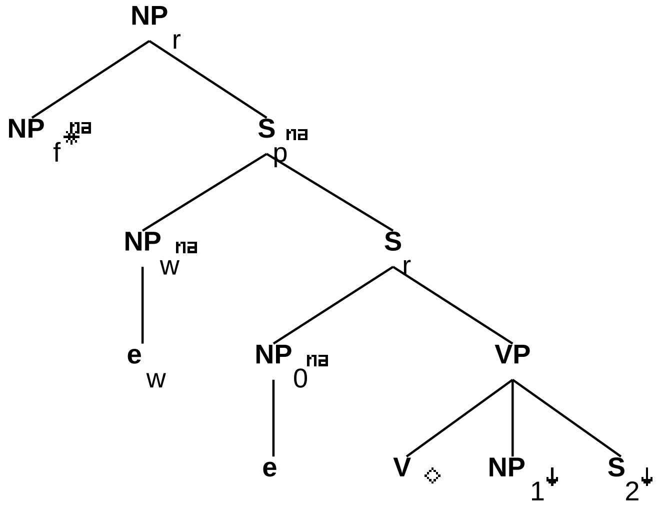
```
Passive tree for verbs taking an NP complement and a
sentential complement. This tree has no by-phrase. The
particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Max was forced to eat spinach.
John was considered to be lazy.
```

26.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.t:<inv> = -
PP_w.t:<wh> = +
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<comp> = nil
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

27 Tree "betaNc0nx0Vnx1s2"

27.1 graphe



27.2 comments

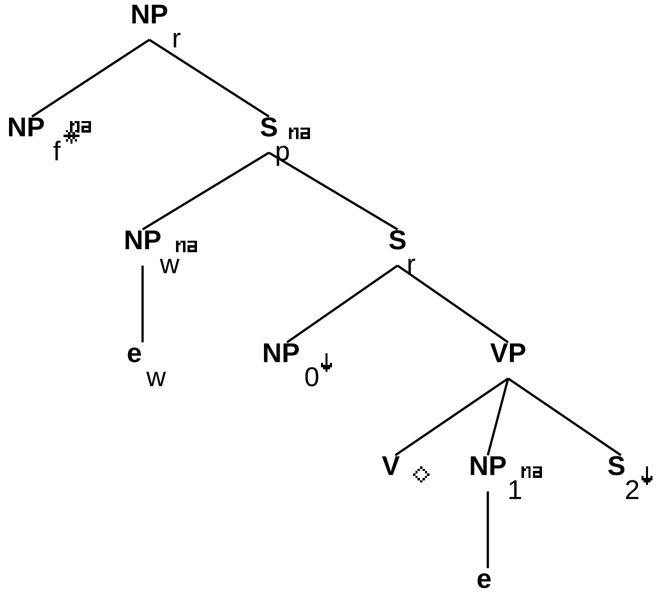
```
Subject relative clause tree for verbs taking an NP complement and
a sentential complement.  The particular verbs place constraints on
the mode of the complement and on which complementizers may adjoin
above the complement.  Note that so-called ECM verbs get this tree
(rather than some sort of small clause analysis).

Exs:  the person who forced Bill to eat spinach
the animal that considers the elephant to be lazy
the fool who bet Mary (that) she could parachute
```

27.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>




S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
S_r.b:<agr> = NP_0:<agr>
S_r.b:<assign-case> = NP_0:<case>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_0.b:<trace>
NP_w.t:<case> = NP_0.b:<case>
NP_w.t:<agr> = NP_0.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind
S_r.t:<nocomp-mode> = inf/ger
VP.t:<assign-comp> = that/ind_nil/inf_nil/ecm
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

28 Tree ”betaNc1nx0Vnx1s2”

28.1 graphe



28.2 comments

Object relative clause tree for verbs taking an NP complement and a sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

Exs: the person who Max forced to eat spinach
the woman Max bet that she could parachute

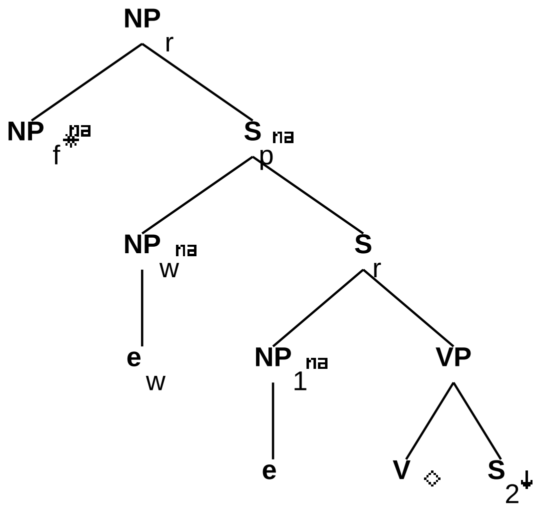
28.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_r.b:<mode> = VP.t:<mode>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_1.t:<case> = acc
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ind
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

29 Tree "betaNc1nx1Vs2"

29.1 graphe



29.2 comments

```
Subject relative clause from passive tree for verbs taking an NP
complement and a sentential complement.  This tree has no by-phrase.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  the emu that was expected to eat oats
the person who was bet that she could not climb Mt. Everest
```

29.3 features

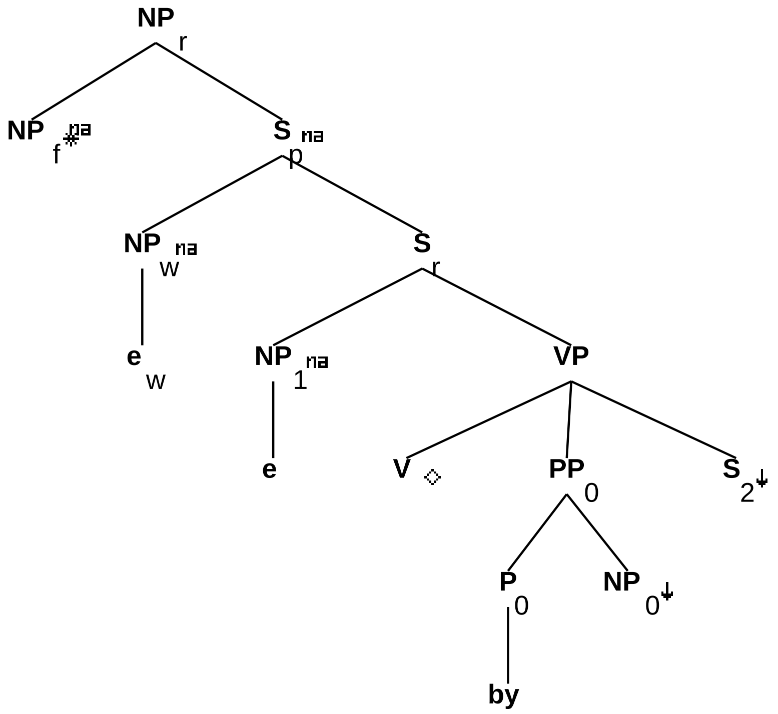
```
NP_r.b:<wh> = NP_f.t:<wh>
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
```

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1.t:<agr> = S_r.b:<agr>
NP_1.t:<case> = S_r.b:<assign-case>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
```

```
VP.t:<passive> = +
VP.b:<mode> = ppart
V.t:<assign-comp> = ppart_nil
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind/ppart
S_r.t:<mode> = ind/inf/ger/ppart
S_r.t:<nocomp-mode> = ind/ger/ppart
VP.t:<assign-comp> = that/inf_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

30 Tree "betaNc1nx1Vbynx0s2"

30.1 graphe



30.2 comments

```
Passive tree for verbs taking an NP
complement and a sentential complement; relative clause on subject.
The by-phrase precedes the sentential complement - there is another
tree with the by-phrase after it.
The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  the person who was forced to eat spinach by Max
the animal that was considered to be lazy by the elephant
```

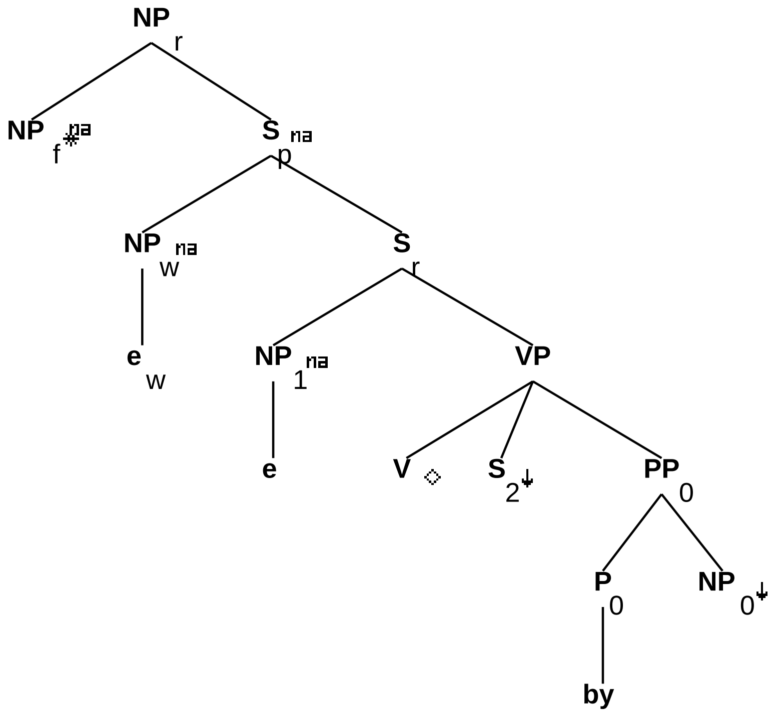
30.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<assign-comp> = ppart_nil
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind/ppart
S_r.t:<nocomp-mode> = ind/ger/ppart
VP.t:<assign-comp> = that/inf_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

31 Tree "betaNc1nx1Vs2bynx0"

31.1 graphe



31.2 comments

```
Passive tree for verbs taking an NP
complement and a sentential complement; relative clause on subject.
The by-phrase follows the sentential complement - there is another
tree with the by-phrase beforee it. The particular verbs place
constraints on the
mode of the complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  the person who was forced by Max to eat spinach
the animal that was considered by the elephant to be lazy
```

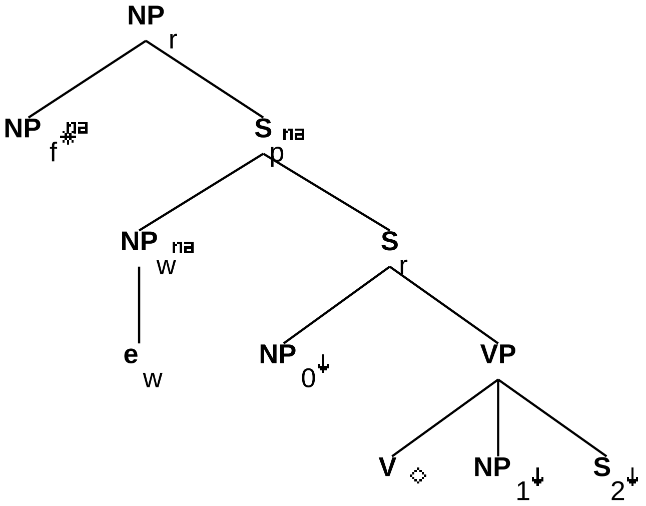
31.3 features

```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_1.t:<agr>
S_r.b:<assign-case> = NP_1.t:<case>
VP.t:<mode> = ind
VP.b:<passive> = +
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
NP_f.b:<refl> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP_1.b:<trace>
NP_w.t:<case> = NP_1.b:<case>
NP_w.t:<agr> = NP_1.b:<agr>
NP_r.b:<rel-clause> = +
S_r.t:<mode> = inf/ger/ind/ppart
S_r.t:<nocomp-mode> = ind/ger/ppart
VP.t:<assign-comp> = that/inf_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>
```

32 Tree ”betaNcnx0Vnx1s2”

32.1 graphe



32.2 comments

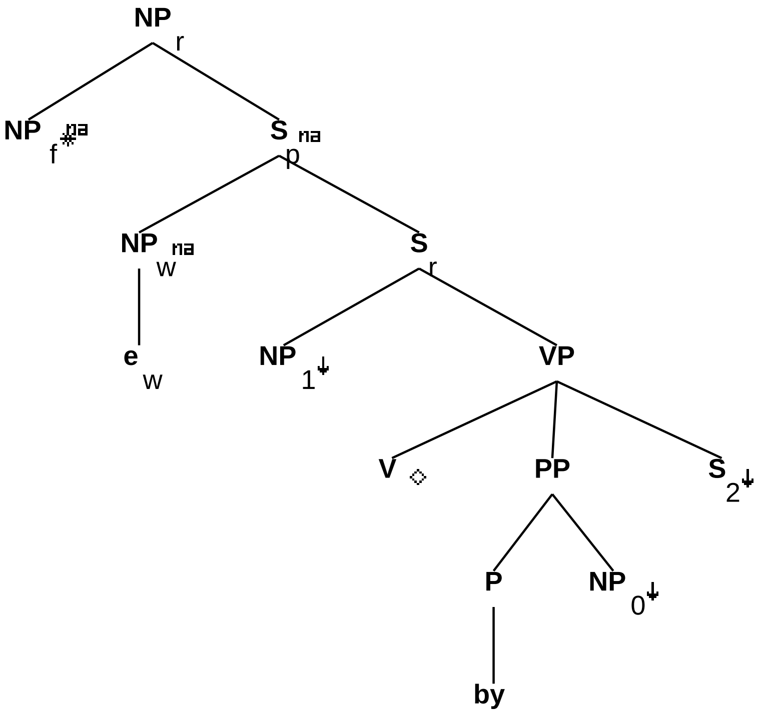
```
Declarative tree for verbs taking an NP complement and a
sentential
complement.  The particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Max forced Bill to eat spinach.
John considers the elephant to be lazy.
John bet Mary (that) she could parachute.
```

32.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>




S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

33 Tree "betaNcnx1Vbynx0s2"

33.1 graphe



33.2 comments

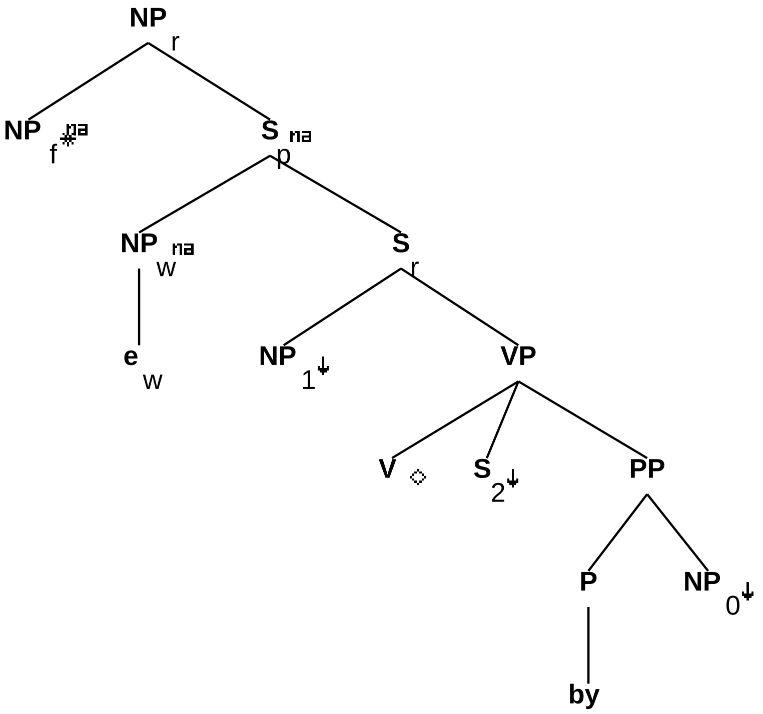
```
Passive tree for verbs taking an NP complement and a
sentential complement.  This tree has the by-phrase before the
sentential complement; there is another tree with the opposite order.
The particular verbs place constraints on the mode of the complement
and on which complementizers may adjoin above the complement.  Note
that so-called ECM verbs get this tree (rather than some sort of small
clause analysis).

Exs:  The emu was expected by his trainer to eat oats.
Mary was bet by Bill that she could not climb Mt. Everest.
```

33.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.t:<mode> = ind
VP.t:<passive> = +
VP.b:<mode> = ppart
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

34 Tree ”betaNcnx1Vs2bynx0”

34.1 graphe



34.2 comments

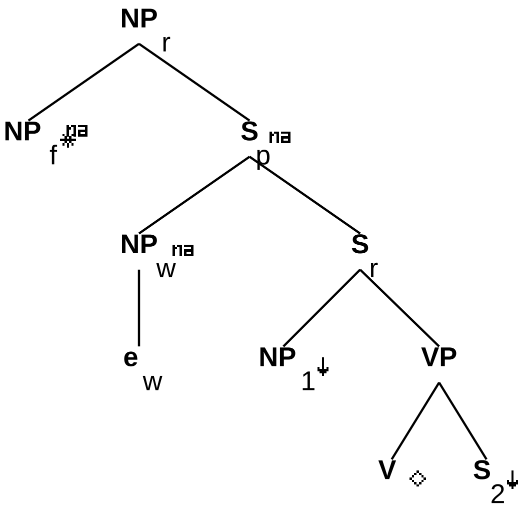
```
Passive tree for verbs taking an NP complement and a
sentential complement. This tree has the by-phrase at the end; there is another
tree with the by-phrase before the sentential complement.  The
particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Max was forced to eat spinach by Bill.
John was considered to be lazy by the elephant.
```

34.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP_0.t:<case>
P.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

35 Tree "betaNcnx1Vs2"

35.1 graphe



35.2 comments

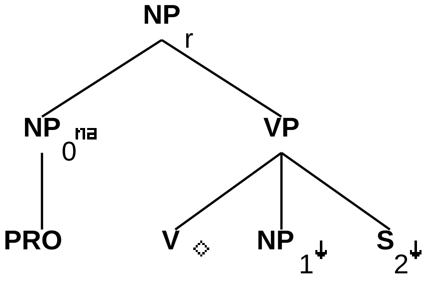
```
Passive tree for verbs taking an NP complement and a
sentential complement. This tree has no by-phrase. The
particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Exs:  Max was forced to eat spinach.
John was considered to be lazy.
```

35.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
NP_1:<wh> = -
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
NP_f.b:<case> = acc/nom
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.t:<nocomp-mode> = ind
VP.t:<assign-comp> = that/for/ind_nil
S_r.b:<nocomp-mode> = S_r.b:<mode>
NP_r.b:<rel-clause> = +
NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
```

36 Tree ”alphaGnx0Vnx1s2-PRO”

36.1 graphe



36.2 comments

NP Gerund tree w/ PRO subject for verbs taking an NP complement and a sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement.

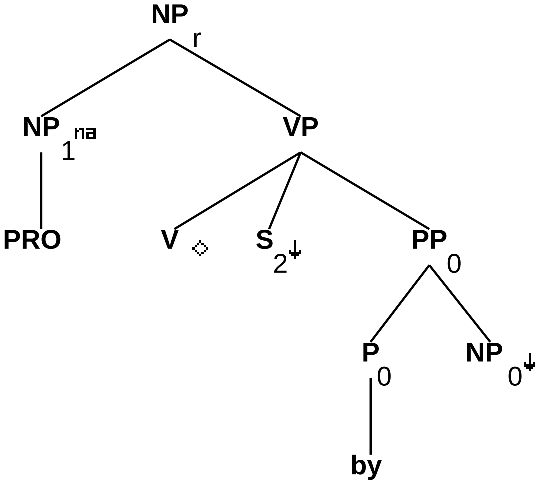
[PRO forcing the emu to dance] was entertaining.

36.3 features

```
NP_0:<wh> = NP_r.b:<wh>
NP_0.t:<wh> = -
NP_0.t:<case> = none
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_1:<case> = acc
VP.t:<mode> = ger

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
NP_r.b:<gerund> = +
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
```

37 Tree "alphaGnx1Vs2bynx0-PRO"

37.1 graphe



37.2 comments

Gerund Passive tree w/ PRO subject for verbs taking an NP complement and a sentential complement, with the 'by' phrase at the end - after the sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement.

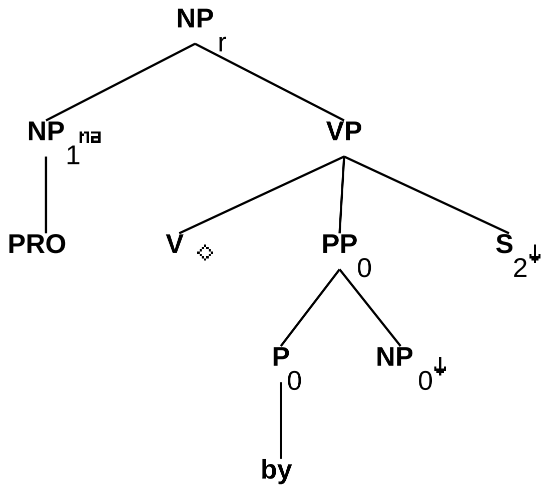
Max dreaded [PRO being forced to eat spinach by Bill].

37.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
NP_1:<wh> = NP_r.b:<wh>
NP_1.t:<wh> = -
NP_1.t:<case> = none
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
```

38 Tree "alphaGnx1Vbynx0s2-PRO"

38.1 graphe



38.2 comments

```
Gerund Passive tree w/ PRO subject for verbs taking an NP complement and a
sentential complement, with the 'by' phrase in the middle - before the
sentential complement. The particular verbs place constraints on the mode of
the complement and on which complementizers may adjoin above the complement.

Max dreaded [PRO being forced by Bill to eat spinach].
```

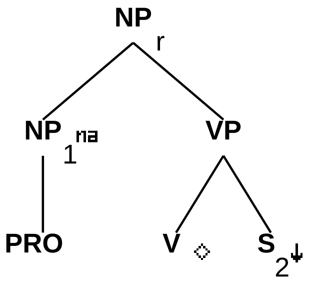
38.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
NP_1:<wh> = NP_r.b:<wh>
NP_1.t:<case> = none
NP_1.t:<wh> = -
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
```

```
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
```

39 Tree ”alphaGnx1Vs2-PRO”

39.1 graphe



39.2 comments

Gerund Passive tree w/ PRO subject for verbs taking an NP complement and a sentential complement, without the ’by’ phrase. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above th complement.

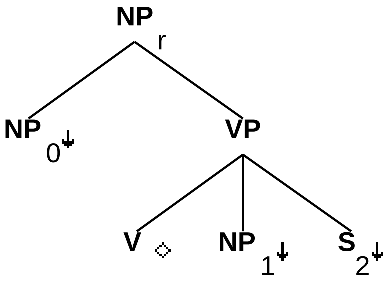
Max dreaded [PRO being forced to eat spinach].

39.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
NP_1:<wh> = NP_r.b:<wh>
NP_1.t:<case> = none
NP_1.t:<wh> = -
VP.t:<mode> = ger
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
```

40 Tree ”alphaGnx0Vnx1s2”

40.1 graphe



40.2 comments

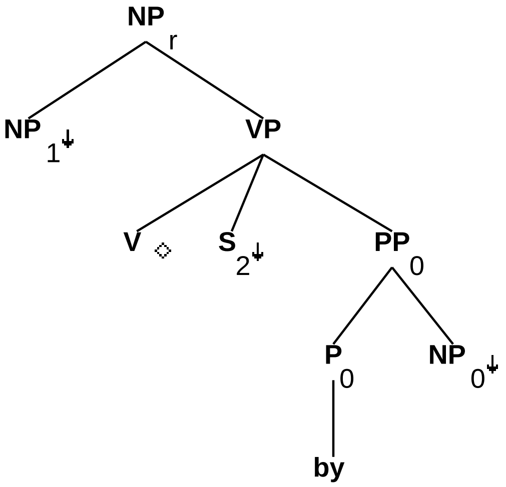
```
NP Gerund tree for verbs taking an NP complement and a
sentential complement. The particular verbs place constraints on the
mode of the complement and on which complementizers may adjoin above
the complement.

Our betting everyone that the Phillies will win (is risky).
Jack betting everyone that the Phillies will win is surprising.
```

40.3 features

```
NP_0:<wh> = NP_r.b:<wh>
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_1:<case> = acc


VP.t:<mode> = ger


S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
NP_r.b:<gerund> = +
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<passive> = -
NP_0:<case> = acc/gen
```

41 Tree "alphaGnx1Vs2bynx0"

41.1 graphe



41.2 comments

```
Gerund Passive tree for verbs taking an NP complement and a sentential
complement, with the 'by' phrase at the end - after the sentential
complement. The particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the complement.

'... 'Max('s) being forced to eat spinach by Bill''
```

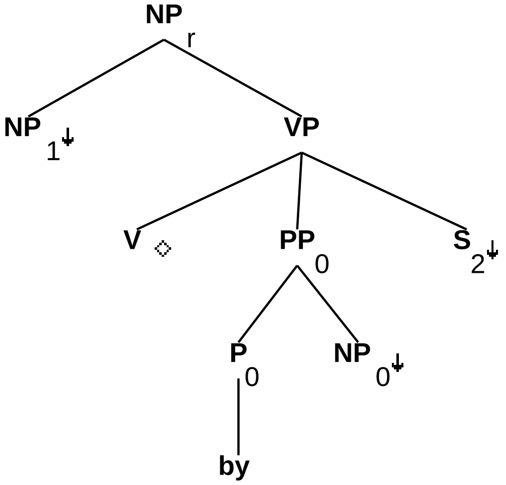
41.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +

NP_r.b:<gerund> = +

NP_1:<wh> = NP_r.b:<wh>
VP.t:<mode> = ger

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
```

```
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
NP_1:<case> = acc/gen
```

42 Tree "alphaGnx1Vbynx0s2"

42.1 graphe



42.2 comments

Gerund Passive tree for verbs taking an NP complement and a sentential complement, with the 'by' phrase in the middle - before the sentential complement. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement.

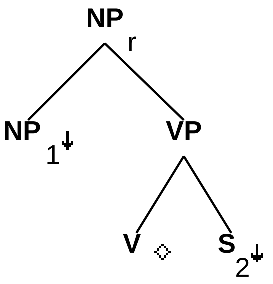
'... 'Max('s) being forced by Bill to eat spinach''

42.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +

NP_r.b:<gerund> = +

NP_1:<wh> = NP_r.b:<wh>
VP.t:<mode> = ger

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
NP_1:<case> = acc/gen
```

43 Tree ”alphaGnx1Vs2”

43.1 graphe



43.2 comments

Gerund Passive tree for verbs taking an NP complement and a sentential complement, without the ’by’ phrase. The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement.

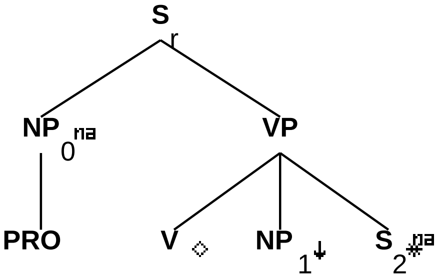
’... ’Max(’s) being forced to eat spinach’’

43.3 features

```
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +

NP_r.b:<gerund> = +

NP_1:<wh> = NP_r.b:<wh>
VP.t:<mode> = ger

VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
NP_1:<case> = acc/gen
```

44 Tree ”betanx0Vnx1s2-PRO”

44.1 graphe



44.2 comments

Declarative tree w/ PRO subject for verbs taking an NP complement and a sentential complem
complement and on which complementizers may adjoin above the complement. Note
that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

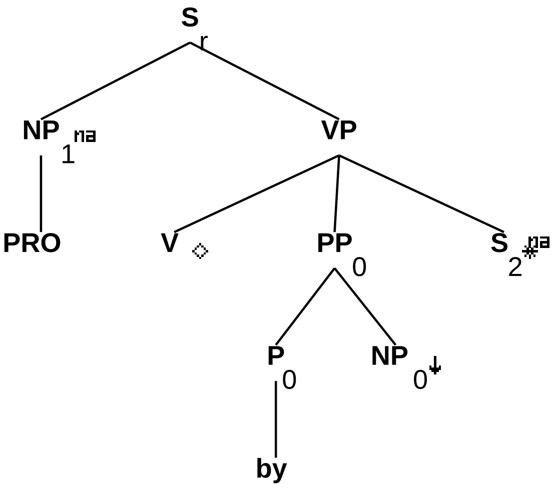
Max wanted [PRO to force Bill to eat spinach].
While [PRO daring Mary to jump off the building] John fell.

44.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = S_r.b:<wh>
NP_0:<wh> = -
NP_0.t:<case> = none
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
VP.b:<agr> = V.t:<agr>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<mode> = V.t:<mode>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_0.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.t:<mode> = inf/ger
```

45 Tree "betanx1Vbynx0s2-PRO"

45.1 graphe



45.2 comments

Passive tree w/ PRO subject for verbs taking an NP complement and a sentential complement. This tree has the by-phrase before the sentential complement; there is anothe
The particular verbs place constraints on the mode of the complement and on which complementizers may adjoin above the complement. Note that so-called ECM verbs get this tree (rather than some sort of small clause analysis).

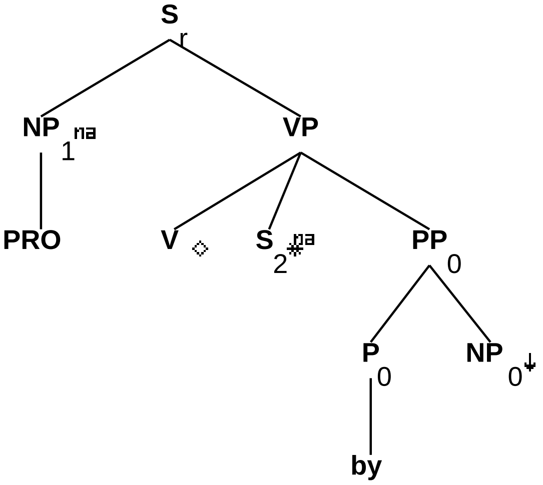
Mary wanted [PRO to be bet by Bill that she could not climb Mt. Everest].
While [PRO being forced by Bill to eat spinach] Mary choked.

45.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = NP_1.t:<case>
NP_1:<agr> = S_r.b:<agr>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
NP_1.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
VP.t:<mode> = ind
VP.t:<passive> = +
VP.b:<mode> = ppart
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.t:<mode> = inf/ger
```

46 Tree "betanx1Vs2bynx0-PRO"

46.1 graphe



46.2 comments

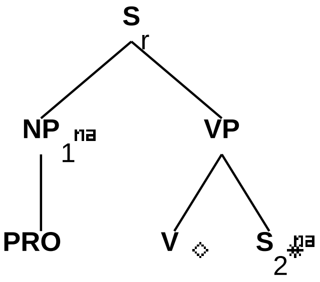
```
Passive tree w/ PRO subject for verbs taking an NP complement and a
sentential complement. This tree has the by-phrase at the end; there is another
tree with the by-phrase before the sentential complement.  The
particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Max wanted [PRO to be forced to eat spinach by Bill].
While [PRO being forced to eat spinach by Bill] Max remembered that he hated spinach.
```

46.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = NP_1.t:<case>
NP_1:<agr> = S_r.b:<agr>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
NP_1.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<inv> = -
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
PP_0.b:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.t:<mode> = inf/ger
```

47 Tree "betanx1Vs2-PRO"

47.1 graphe



47.2 comments

```
Passive tree w/ PRO subject for verbs taking an NP complement and a
sentential complement. This tree has no by-phrase. The
particular verbs place constraints on the mode of the
complement and on which complementizers may adjoin above the
complement.  Note that so-called ECM verbs get this tree (rather than
some sort of small clause analysis).

Max wanted [PRO to be forced to eat spinach].
While [PRO telling John that it was her turn] Mary realized that it was his turn.
```

47.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = NP_1.t:<case>
NP_1:<agr> = S_r.b:<agr>
NP_1:<wh> = S_r.b:<wh>
NP_1:<wh> = -
NP_1.t:<case> = none
S_r.b:<agr> = VP.t:<agr>
VP.b:<mode> = V.t:<mode>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<agr> = V.t:<agr>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
V.t:<mode> = ppart
V.t:<passive> = +

S_2.t:<assign-comp> = inf_nil/ind_nil
S_2.t:<inv> = -
S_r.b:<control> = NP_1.t:<control>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<mainv> = VP.t:<mainv>
VP.t:<mode> = inf/ger
```