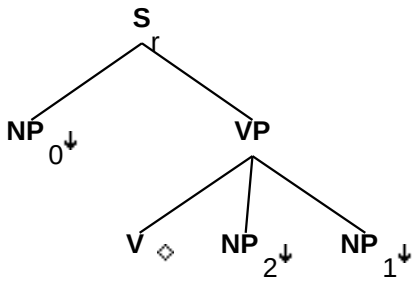
Family "Tnx0Vnx2nx1"

March 5, 2008

1 Tree "alphanx0Vnx2nx1"

1.1 graphe



1.2 comments

no comments

1.3 features

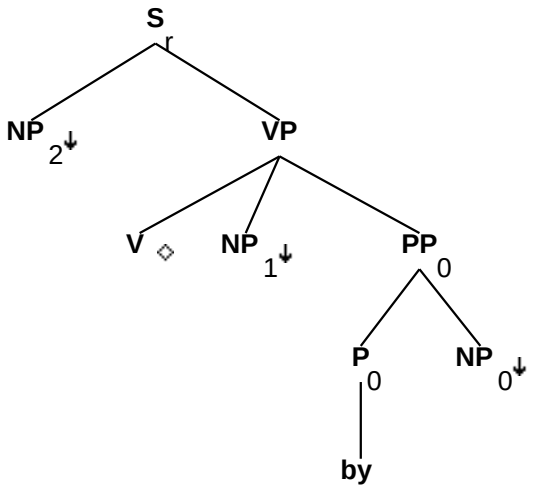
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_0:<wh>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<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_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 "alphanx2Vnx1bynx0"

2.1 graphe



2.2 comments

Passive tree:
Mary was asked a question by John.

2.3 features

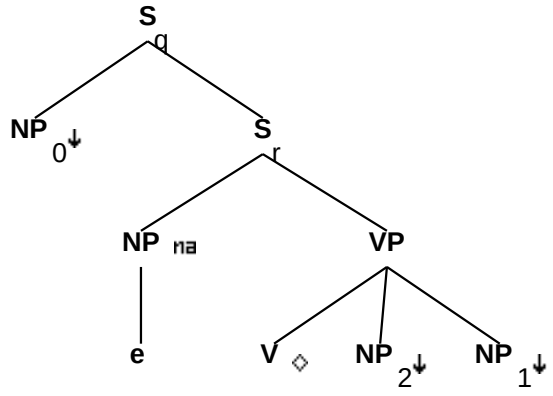
S_r.b:<inv> = -
S_r.b:<extracted> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart

```
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
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>
P_0.b:<assign-case> = acc
S_r.b:<control> = NP_2.t:<control>
PP_0.b:<wh> = NP_0:<wh>
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>
```

3 Tree "alphaW0nx0Vnx2nx1"

3.1 graphe



3.2 comments

Need to decide what VP agrees with.

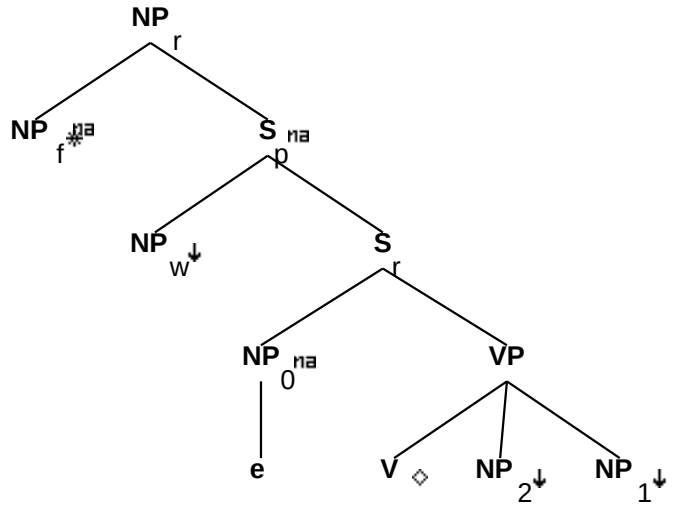
3.3 features

```
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_q.b:<wh> = NP_0.t:<wh>
S_q.b:<comp> = nil
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<inv> = S_r.t:<inv>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<case>
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> = +
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_1:<case> = acc
NP_2:<case> = acc
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_r.t:<conj> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
```

4 Tree "betaN0nx0Vnx2nx1"

4.1 graphe



4.2 comments

Need to decide what VP agrees with.

4.3 features

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

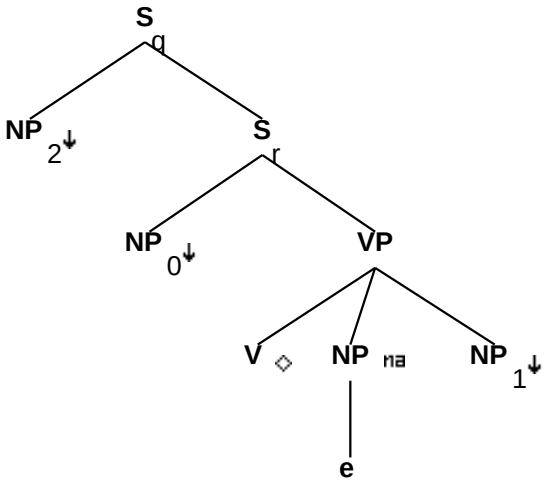
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.t:<agr> = NP_r.b:<agr>
S_r.b:<comp> = nil
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>

5 Tree "alphaW2nx0Vnx2nx1"

5.1 graphe



5.2 comments

No original comments.

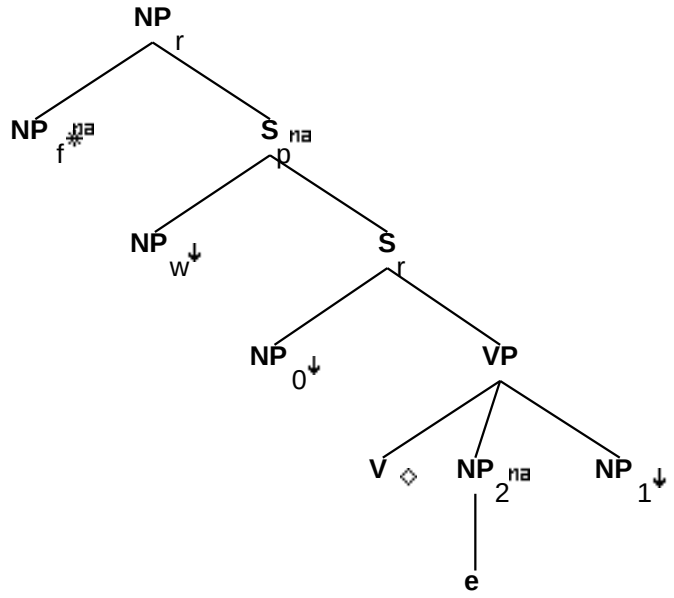
5.3 features

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

```
S_q.b:<wh> = NP_2:<wh>
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<inv> = S_r.t:<inv>
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.t:<case> = acc
NP.t:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_2:<trace> = NP.t:<trace>
NP_2:<agr> = NP.t:<agr>
NP_2:<case> = NP.t:<case>
NP_2:<wh> = NP.t:<wh>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<control> = NP_0.t:<control>
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_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>
```

6 Tree "betaN1nx0Vnx2nx1"

6.1 graphe



6.2 comments

No original comments.

6.3 features

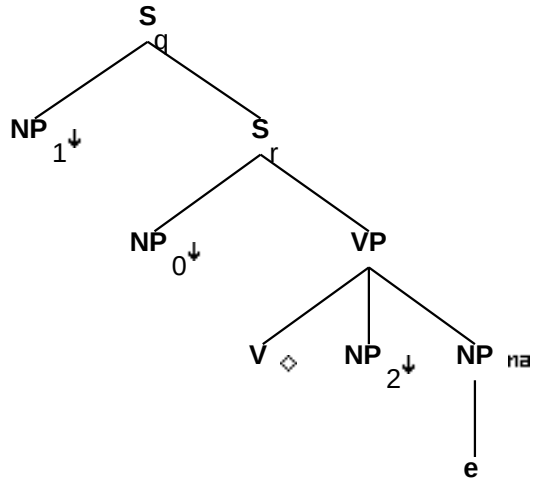
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = VP.t:<mode>
S_r.t:<inv> = -
S_r.t:<mode> = ind/inf
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_2:<case> = acc
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-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>
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_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>
```

7 Tree "alphaW1nx0Vnx2nx1"

7.1 graphe



7.2 comments

No original comments.

7.3 features

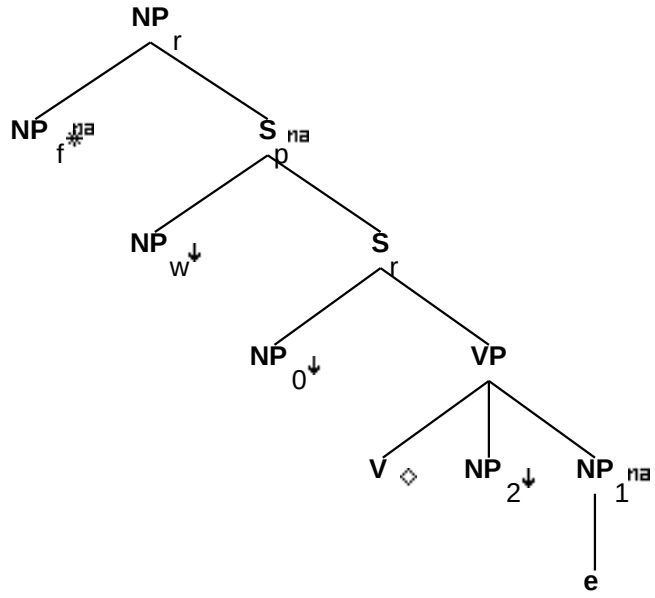
```
S_q.b:<extracted> = +
S_r.t:<comp> = nil
```

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

```
S_q.b:<wh> = NP_1.t:<wh>
S_q.b:<inv> = S_q.b:<invlink>
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.t:<inv> = S_q.b:<inv>
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_2:<case> = acc
NP.t:<case> = acc
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:<agr> = NP_1.t:<agr>
NP.t:<case> = NP_1.t:<case>
NP.t:<wh> = NP_1.t:<wh>
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_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>
```

8 Tree "betaN2nx0Vnx2nx1"

8.1 graphe



8.2 comments

No original comments.

8.3 features

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

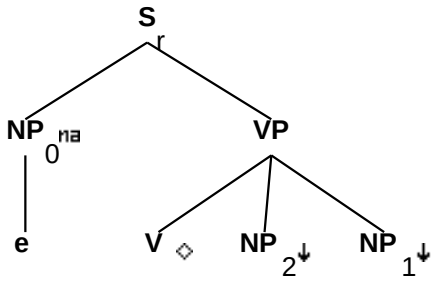
S_r.b:<mode> = VP.t:<mode>
S_r.t:<mode> = ind/inf
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<case> = acc
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_r.b:<wh> = NP_f.t:<wh>
NP_r.b:<agr> = NP_f.t:<agr>
NP_r.b:<case> = NP_f.t:<case>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<control> = NP_0.t:<control>

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_r.t:<conj> = nil

NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.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>

9 Tree "alphaInx0Vnx2nx1"

9.1 graphe



9.2 comments

no comments

9.3 features

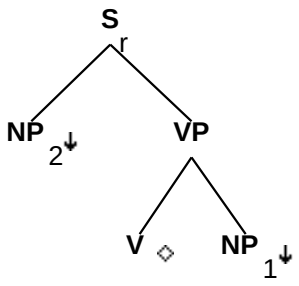
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_r.b:<mode> = imp
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>

```
S_r.b:<wh> = NP_0:<wh>
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
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
NP_2:<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_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 "alphanx2Vnx1"

10.1 graphe



10.2 comments

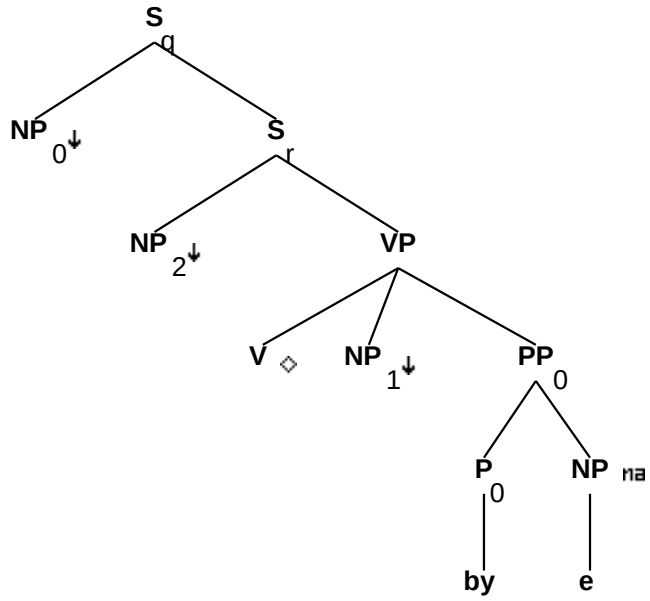
```
Passive w/out by-phrase:
Mary was asked a question.
```

10.3 features

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



VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_r.b:<control> = NP_2.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>
```

11 Tree "alphaW0nx2Vnx1bynx0"

11.1 graphe



11.2 comments

passive, extraction from by-phrase:

who was Mary asked a question by

11.3 features

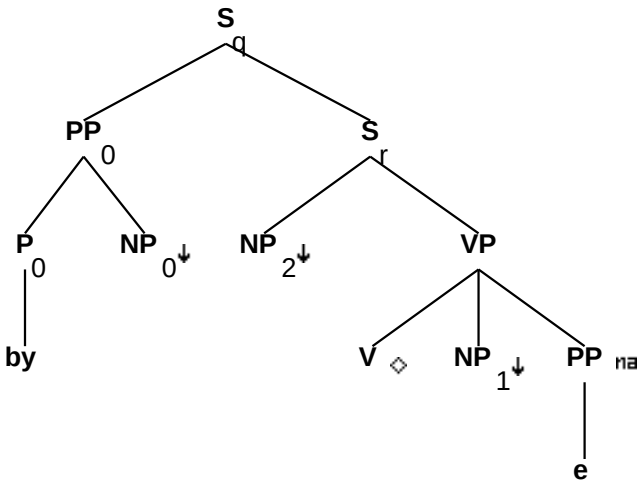
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_q.b:<wh> = NP_0:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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:<agr> = NP_0.t:<agr>
NP.t:<case> = NP_0.t:<case>
NP.t:<trace> = NP_0.t:<trace>
NP.t:<wh> = NP_0.t:<wh>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP:<case> = PP_0.b:<assign-case>
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.t:<control>
PP_0.b:<wh> = NP:<wh>
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>
```

12 Tree "alphapW0nx2Vnx1bynx0"

12.1 graphe



12.2 comments

passive, extraction of by-phrase:

by whom was Mary asked a question

12.3 features

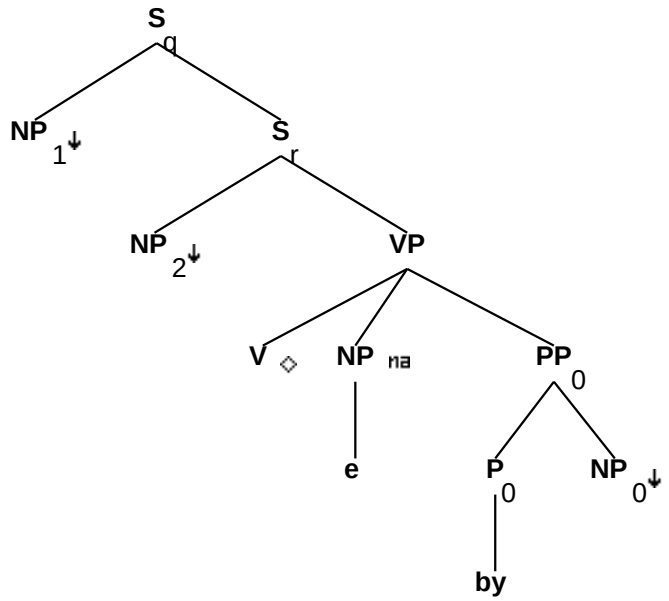
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>

S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = PP_0.t:<wh>
PP_0.b:<wh> = NP_0:<wh>
S_q.b:<inv> = S_r.t:<inv>
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_2:<agr> = VP.t:<agr>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP_0:<case> = PP_0.b:<assign-case>
NP_1:<case> = acc
PP_0:<trace> = PP.t:<trace>
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.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>
```

13 Tree "alphaW1nx2Vnx1bynx0"

13.1 graphe



13.2 comments

```
passive, extraction of NP2:
what was Mary asked by Max
```

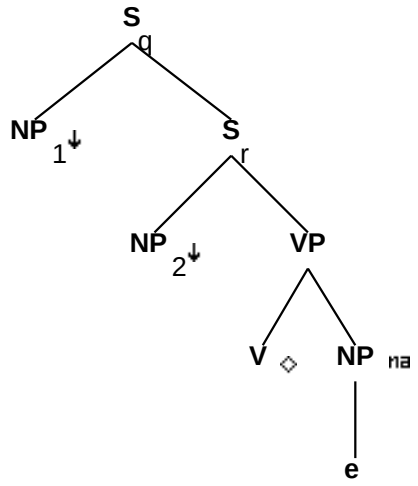
13.3 features

```
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



S_q.b:<inv> = S_q.b:<invlink>

S_q.b:<wh> = NP_1:<wh>
S_q.b:<inv> = S_r.t:<inv>
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_1:<agr> = NP.t:<agr>
NP_1:<case> = NP.t:<case>
NP_1:<trace> = NP.t:<trace>
NP_1:<wh> = NP.t:<wh>
P_0.b:<assign-case> = acc
PP_0.b:<assign-case> = P_0.t:<assign-case>
NP_0:<case> = PP_0.b:<assign-case>
NP:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.t:<control>
PP_0.b:<wh> = NP_0:<wh>
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 "alphaW1nx2Vnx1"

14.1 graphe



14.2 comments

```
passive, extraction of NP2, w/o by-phrase:
what was Mary asked
```

14.3 features

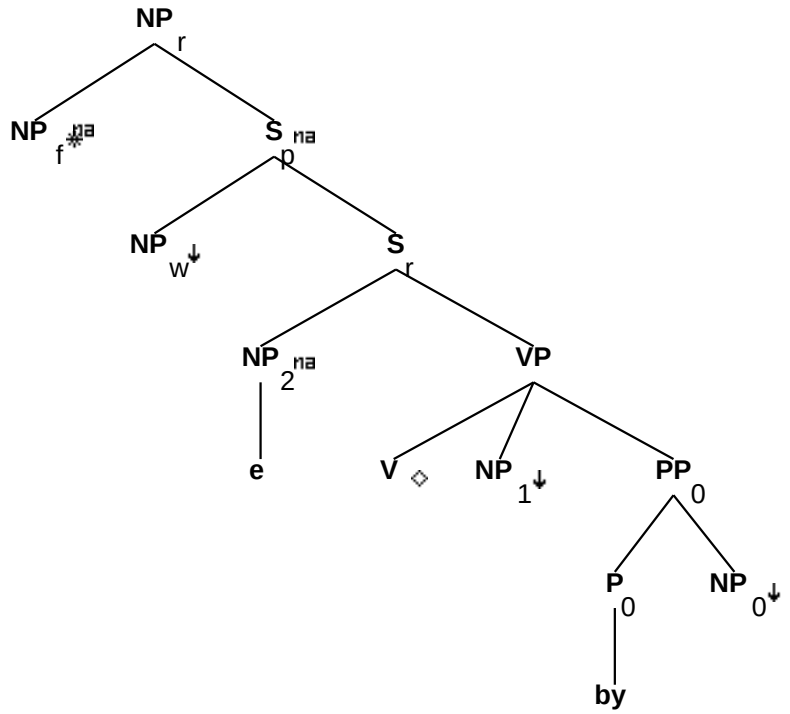
```
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<wh> = NP_1:<wh>
S_q.b:<inv> = S_r.t:<inv>
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_1:<agr> = NP.t:<agr>
NP_1:<case> = NP.t:<case>
NP_1:<trace> = NP.t:<trace>
NP_1:<wh> = NP.t:<wh>
NP:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.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>

15 Tree "betaN1nx2Vnx1bynx0"

15.1 graphe



15.2 comments

```
That relative clause, extraction from NP1:
(I saw) the person that the question was asked by Max
```

15.3 features

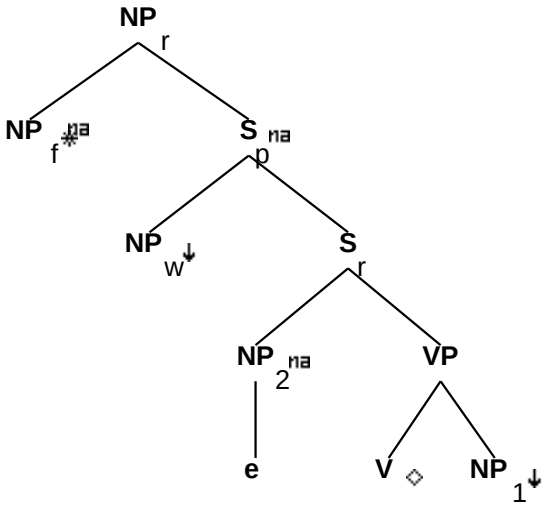
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
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
PP_0.b:<wh> = NP_0:<wh>
```

NP_r.b:<pron> = NP_f.t:<pron>

16 Tree "betaN1nx2Vnx1"

16.1 graphe



16.2 comments

Passive that relative clause, extraction from NP1, w/o by-phrase:
(I saw) the person that the question was asked

16.3 features

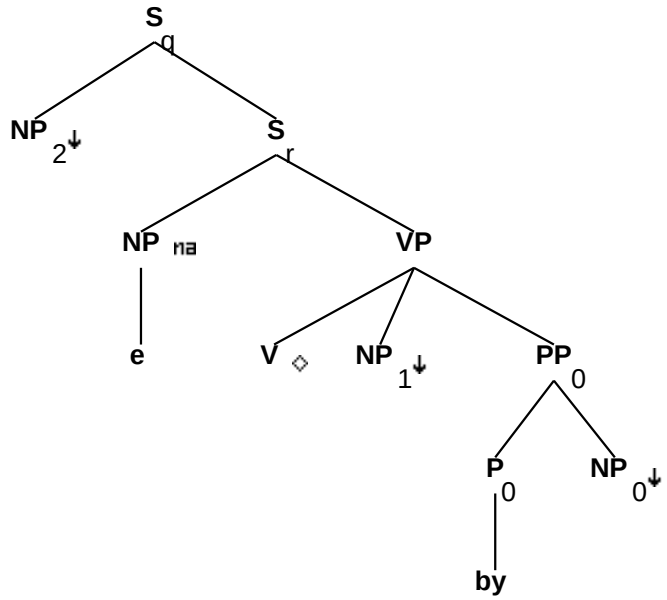
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>

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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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> = -
NP_1:<case> = acc
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>
```

17 Tree "alphaW2nx2Vnx1bynx0"

17.1 graphe



17.2 comments

Wh question on NP1 in passive constructions

who was asked a question by Max

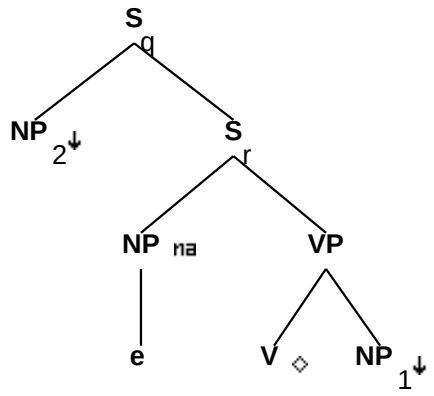
17.3 features

S_q.b:<extracted> = +
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_q.b:<wh> = NP_2.t:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
NP_2:<agr> = NP.t:<agr>
NP_2:<case> = NP.t:<case>
NP.t:<wh> = +
NP_2:<trace> = NP.t:<trace>
NP_2:<wh> = NP.t:<wh>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<inv> = -
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.t:<agr>
S_r.b:<assign-case> = NP.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>
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
S_r.t:<conj> = nil
PP_0.b:<wh> = NP_0:<wh>
NP_1.t:<case> = acc

18 Tree "alphaW2nx2Vnx1"

18.1 graphe



18.2 comments

```
Wh question on NP2 in passive constructions, w/o by-phrase

what was asked Mary
```

18.3 features

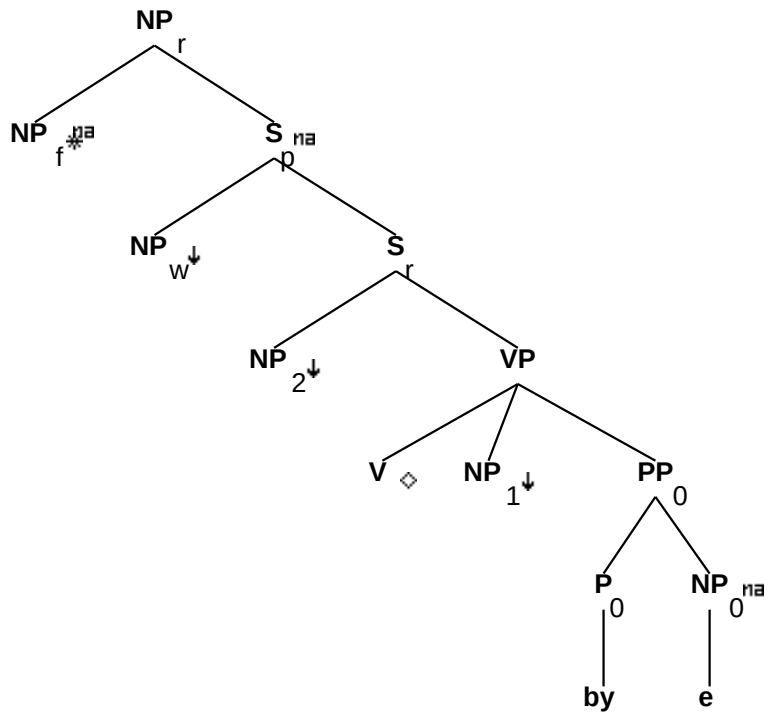
```
S_q.b:<extracted> = +
S_r.t:<comp> = nil
S_r.b:<assign-comp> = inf_nil/ind_nil/ecm
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_q.b:<wh> = NP_2:<wh>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
NP_2:<agr> = NP.t:<agr>
NP_2:<case> = NP.t:<case>
NP.t:<wh> = +
NP_2:<trace> = NP.t:<trace>
NP_2:<wh> = NP.t:<wh>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<inv> = -
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.t:<agr>
```

S_r.b:<assign-case> = NP.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_1:<case> = acc
S_r.t:<conj> = nil

19 Tree "betaN0nx2Vnx1bynx0"

19.1 graphe



19.2 comments

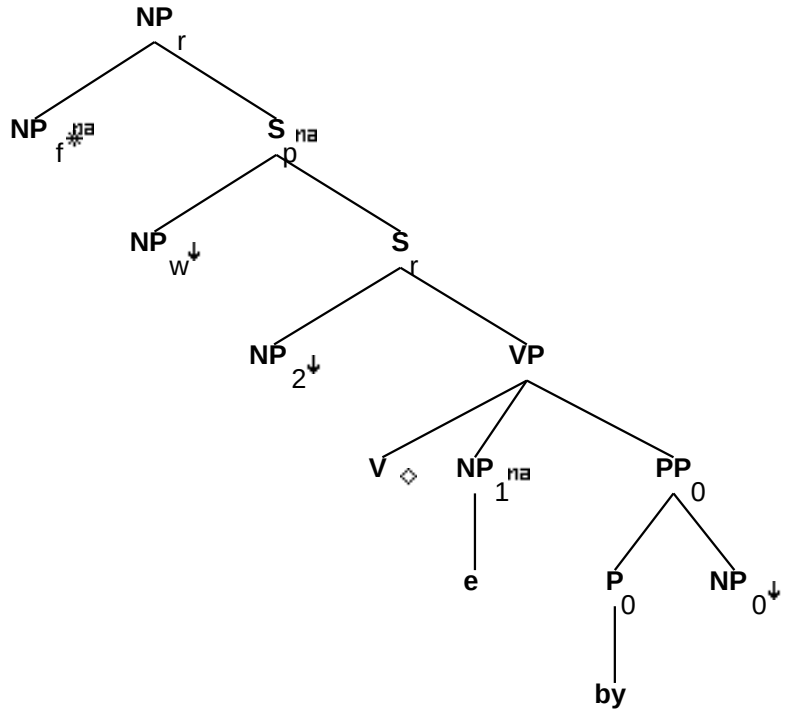
That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that Mary was asked the question by

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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>

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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

20 Tree "betaN2nx2Vnx1bynx0"

20.1 graphe



20.2 comments

```
That relative clause, extraction from NP2:
(I know) the question that Mary was asked by Max
```

20.3 features

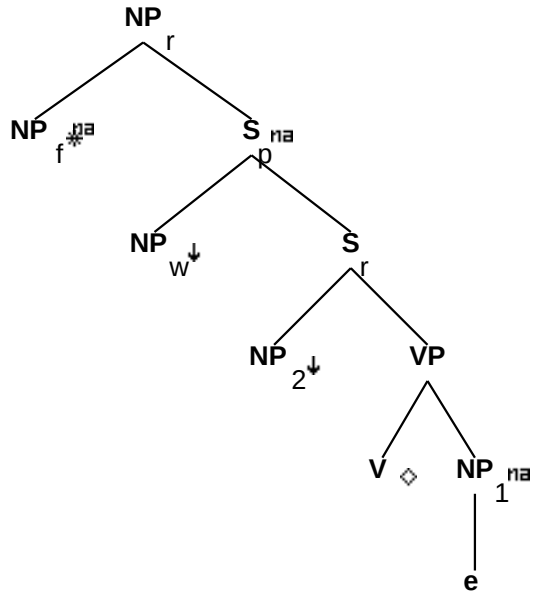
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.t:<agr>
```

```
S_r.b:<assign-case> = NP_2.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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

21 Tree "betaN2nx2Vnx1"

21.1 graphe



21.2 comments

```
That relative clause, extraction from NP2, w/o by-phrase:
(I know) the question that Mary was asked
```

21.3 features

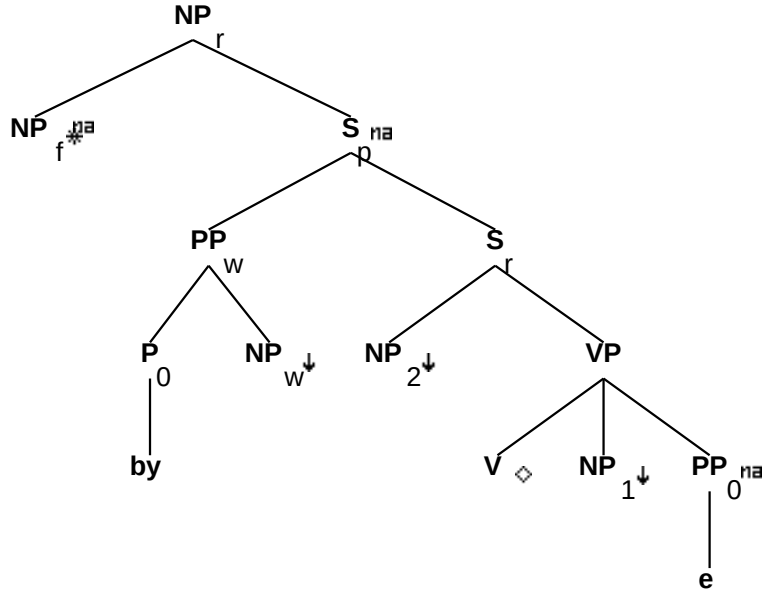
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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> = -
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.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>
```

22 Tree "betaNbynx0nx2Vnx1bynx0"

22.1 graphe



22.2 comments

```
That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that the person that Mary was asked the question by
```

22.3 features

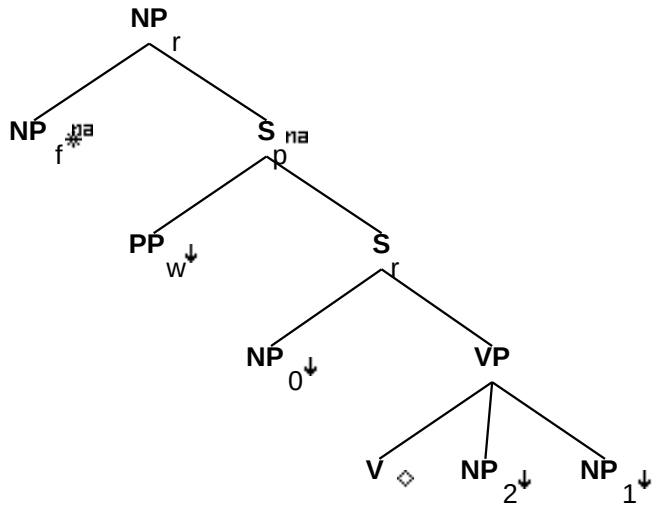
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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> = -
P_0.b:<assign-case> = acc
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_2.t:<control>

NP_w.t:<wh> = +
S_r.t:<comp> = nil
PP_w.t:<trace> = PP_0.b:<trace>
PP_w.t:<case> = PP_0.b:<case>
PP_w.t:<agr> = PP_0.b:<agr>
PP_w.b:<assign-case> = P_0.t:<assign-case>
PP_w.b:<assign-case> = NP_w.t:<case>
PP_w.b:<wh> = NP_w.t:<wh>
NP_r.b:<rel-clause> = +
```

NP_f.b:<case> = nom/acc
NP_r.b:<pron> = NP_f.t:<pron>

23 Tree "betaNpxnx0Vnx2nx1"

23.1 graphe



23.2 comments

no comments

23.3 features

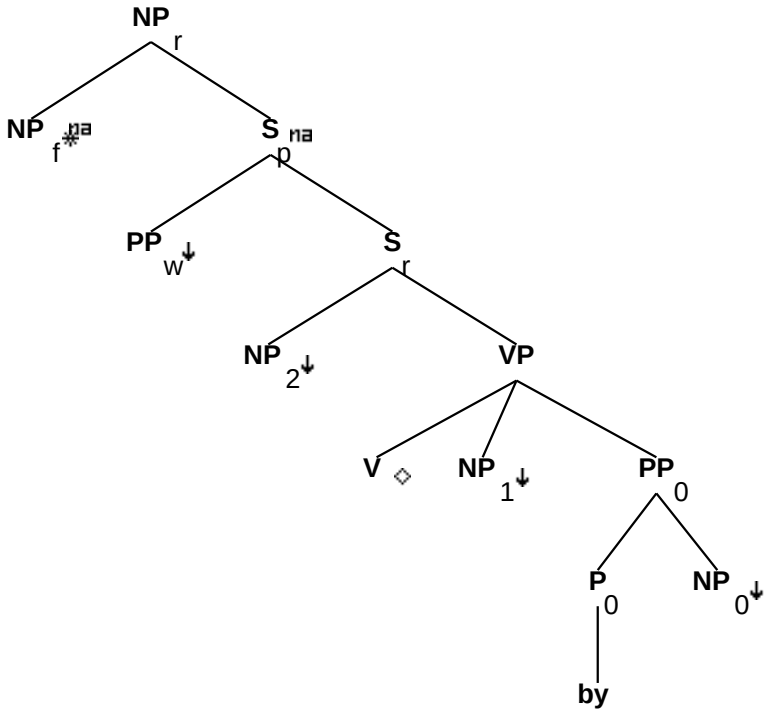
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>

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
NP_2:<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_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 "betaNpxnx2Vnx1bynx0"

24.1 graphe



24.2 comments

```
Passive tree:
Mary was asked a question by John.
```

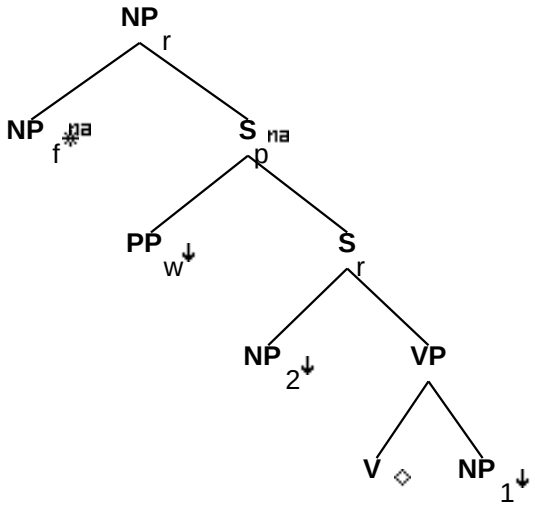
24.3 features

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



VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
```

```
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
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>
P_0.b:<assign-case> = acc
S_r.b:<control> = NP_2.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
PP_0.b:<wh> = NP_0:<wh>
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 "betaNpxnx2Vnx1"

25.1 graphe



25.2 comments

```
Passive w/out by-phrase:
Mary was asked a question.
```

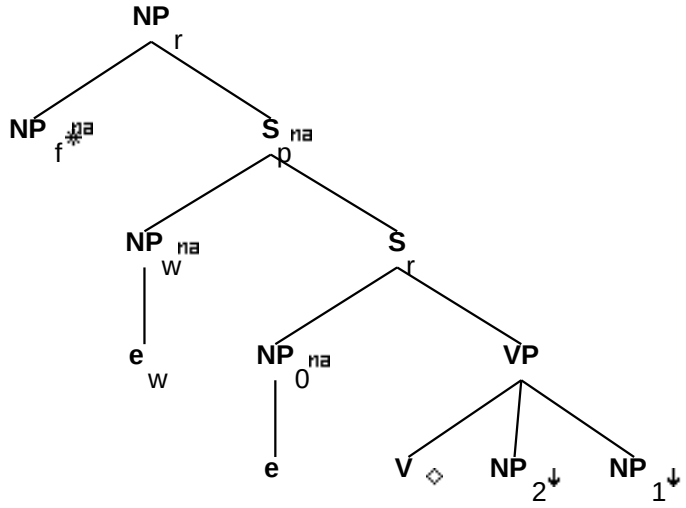
25.3 features

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



VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
```

```
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
S_r.b:<control> = NP_2.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 "betaNc0nx0Vnx2nx1"

26.1 graphe



26.2 comments

```
Need to decide what VP agrees with.
```

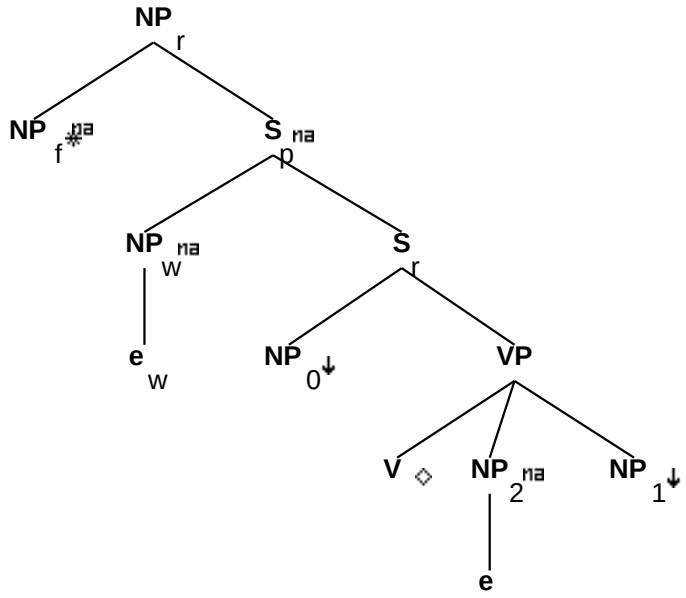
26.3 features

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

```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<case>
S_r.t:<inv> = -
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
NP_f.t:<agr> = NP_r.b:<agr>
S_r.b:<comp> = nil
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
NP_1:<case> = acc
NP_2:<case> = acc
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_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>
```

27 Tree "betaNc1nx0Vnx2nx1"

27.1 graphe



27.2 comments

No original comments.

27.3 features

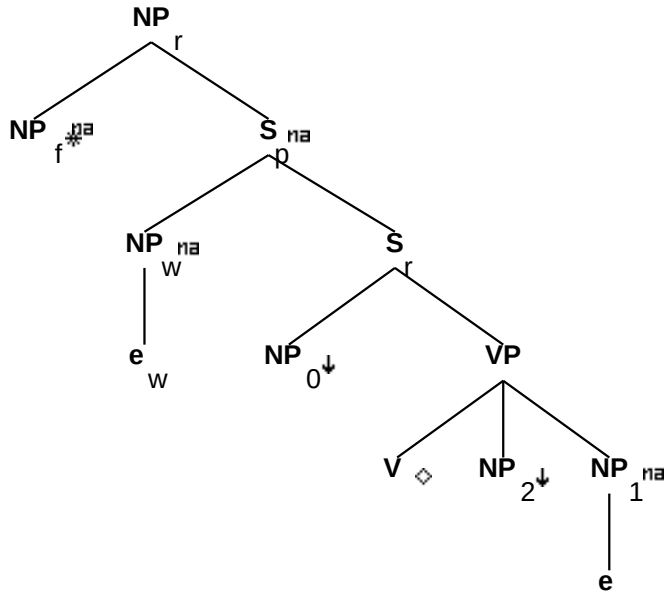
S_r.b:<assign-comp> = VP.t:<assign-comp>

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>
NP_2:<case> = acc
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-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>
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_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>
```

28 Tree "betaNc2nx0Vnx2nx1"

28.1 graphe



28.2 comments

No original comments.

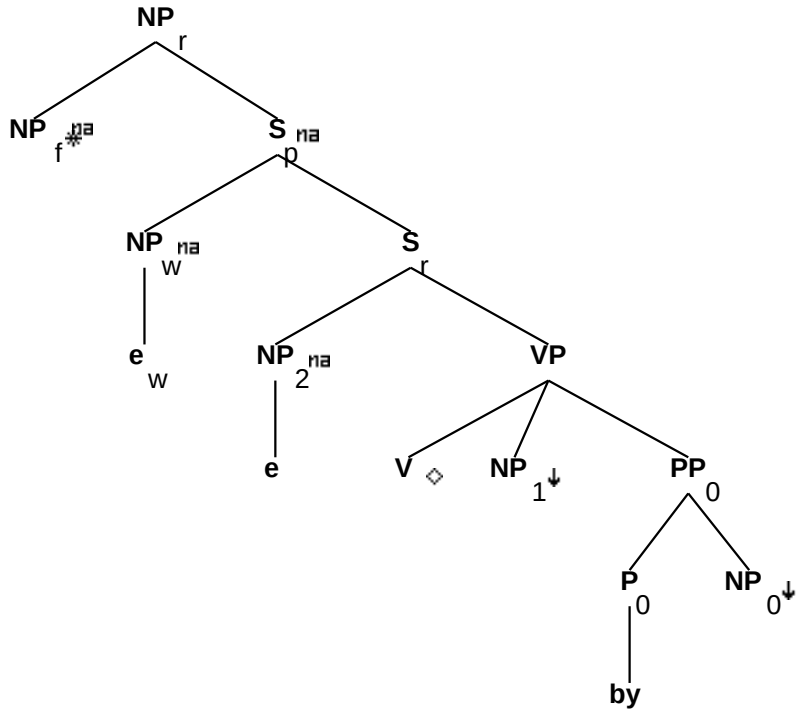
28.3 features

```
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>
NP_1:<case> = acc
NP_2:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-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>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<control> = NP_0.t:<control>
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_r.t:<conj> = nil

NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.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 "betaNc1nx2Vnx1bynx0"

29.1 graphe



29.2 comments

```
That relative clause, extraction from NP1:
(I saw) the person that was asked a question by Max
```

29.3 features

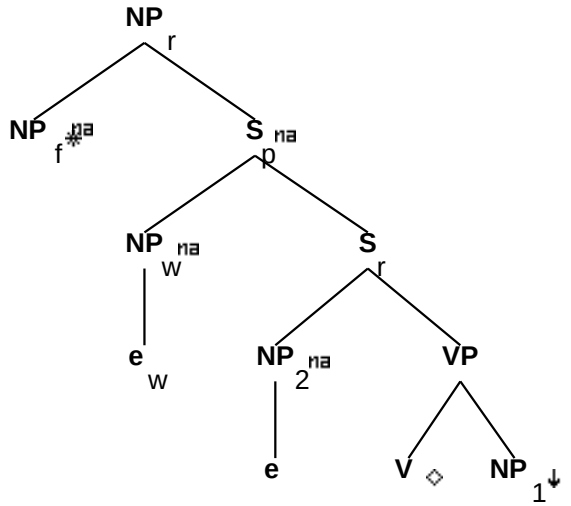
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

30 Tree "betaNc1nx2Vnx1"

30.1 graphe



30.2 comments

```
Passive that relative clause, extraction from NP1, w/o by-phrase:
(I saw) the person that was asked a question
```

30.3 features

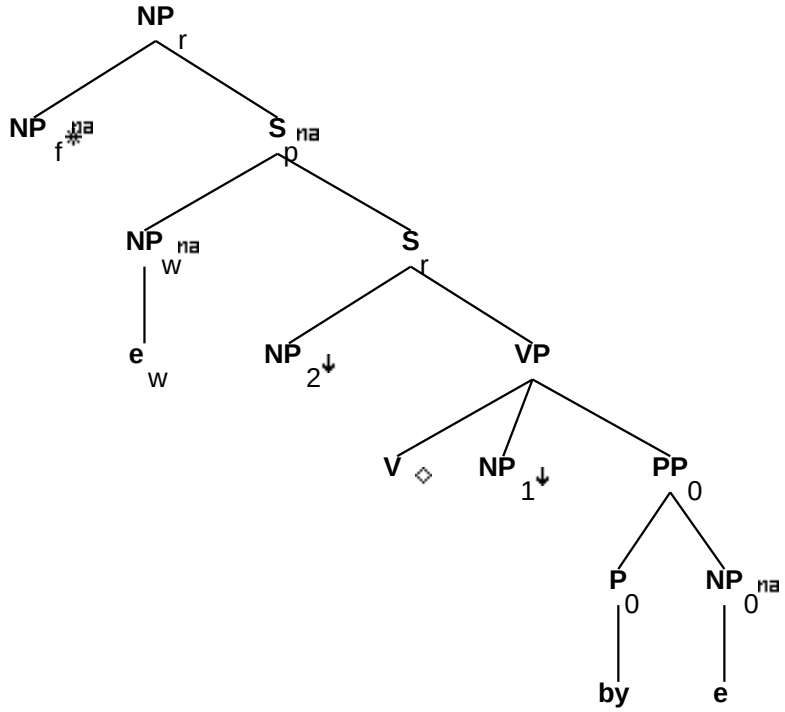
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.t:<agr>
S_r.b:<assign-case> = NP_2.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> = -
NP_1:<case> = acc
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 "betaNc0nx2Vnx1bynx0"

31.1 graphe



31.2 comments

That relative clause, extraction of NP0 from by-phrase:
(I saw) the person that the person that Mary was asked the question by

31.3 features

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

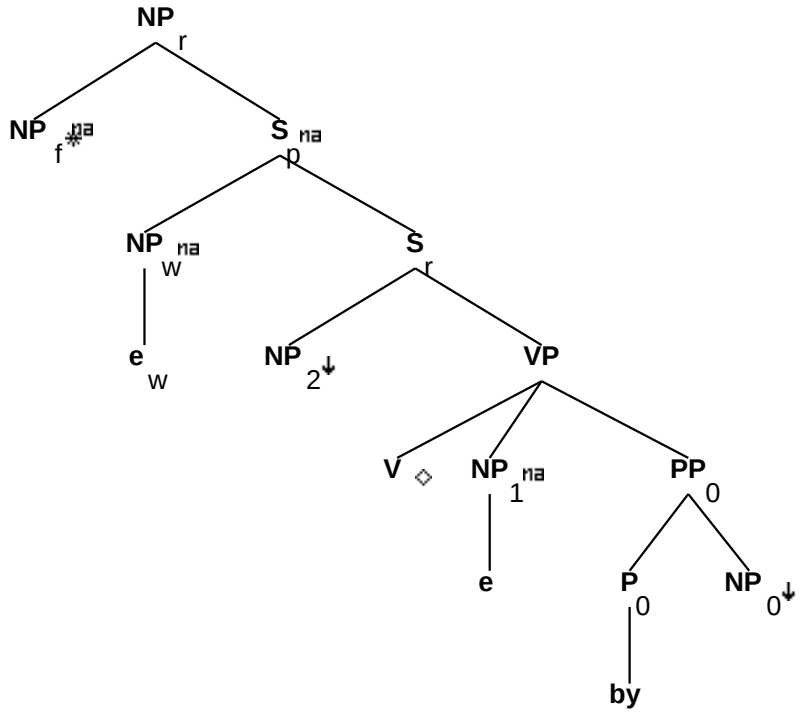
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
S_r.b:<comp> = nil
S_r.b:<mode> = VP.t:<mode>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<agr> = NP_2.t:<agr>
S_r.b:<assign-case> = NP_2.t:<case>
VP.t:<mode> = 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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>

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/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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

32 Tree "betaNc2nx2Vnx1bynx0"

32.1 graphe



32.2 comments

```
That relative clause, extraction from NP2:
(I know) the question that Mary was asked by Max
```

32.3 features

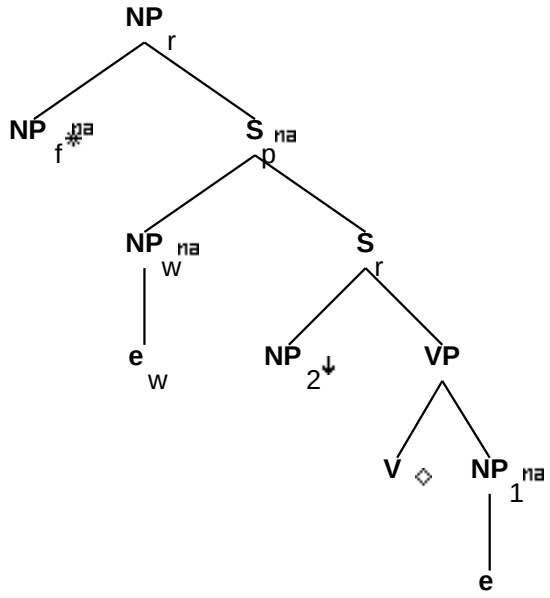
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.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_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP_0.t:<case>
P_0.b:<assign-case> = acc
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>

NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.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
PP_0.b:<wh> = NP_0:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

33 Tree "betaNc2nx2Vnx1"

33.1 graphe



33.2 comments

```
That relative clause, extraction from NP2, w/o by-phrase:
(I know) the question that Mary was asked
```

33.3 features

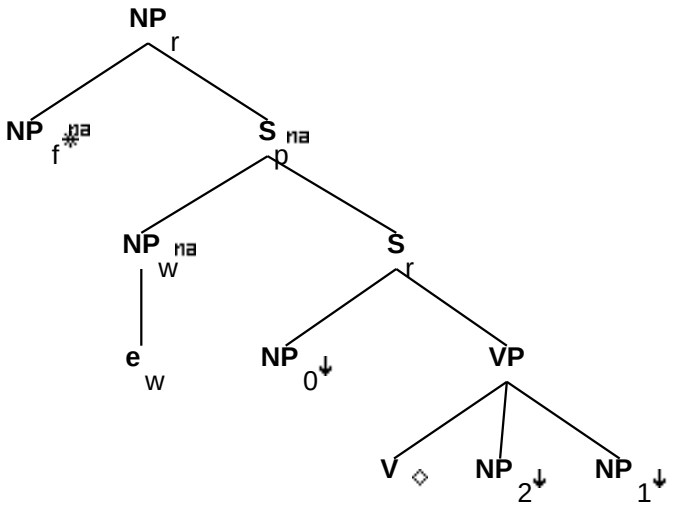
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
NP_f.t:<agr> = NP_r.b:<agr>
NP_f.t:<wh> = NP_r.b:<wh>
NP_f.t:<case> = NP_r.b:<case>
```

```
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_2.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> = -
NP_1:<case> = acc
S_r.t:<conj> = nil
S_r.b:<control> = NP_1.t:<control>

NP_w.t:<trace> = NP_2.b:<trace>
NP_w.t:<case> = NP_2.b:<case>
NP_w.t:<agr> = NP_2.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>
```

34 Tree "betaNcnx0Vnx2nx1"

34.1 graphe



34.2 comments

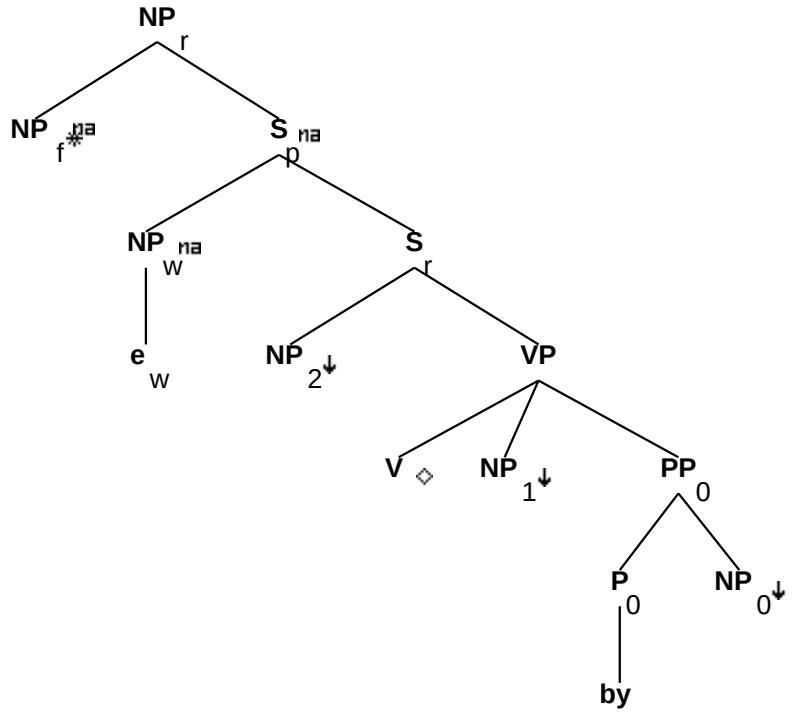
no comments

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_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_0:<wh> = -
NP_1:<case> = acc
NP_2:<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_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>
```

35 Tree "betaNcnx2Vnx1bynx0"

35.1 graphe



35.2 comments

```
Passive tree:
Mary was asked a question by John.
```

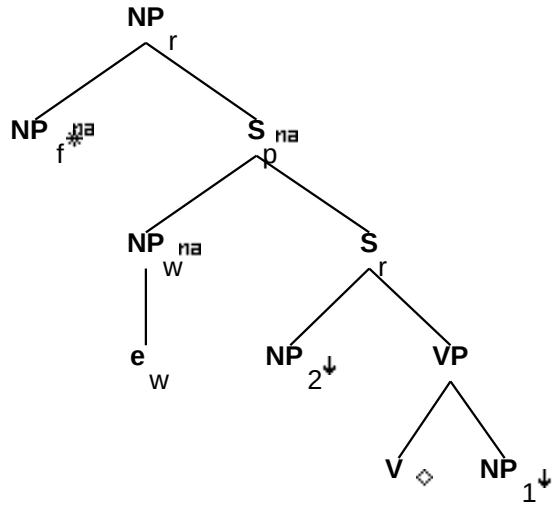
35.3 features

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



VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
```

```
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
VP.b:<assign-comp> = V.t:<assign-comp>
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>
P_0.b:<assign-case> = acc
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
PP_0.b:<wh> = NP_0:<wh>
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 "betaNcnx2Vnx1"

36.1 graphe



36.2 comments

```
Passive w/out by-phrase:
Mary was asked a question.
```

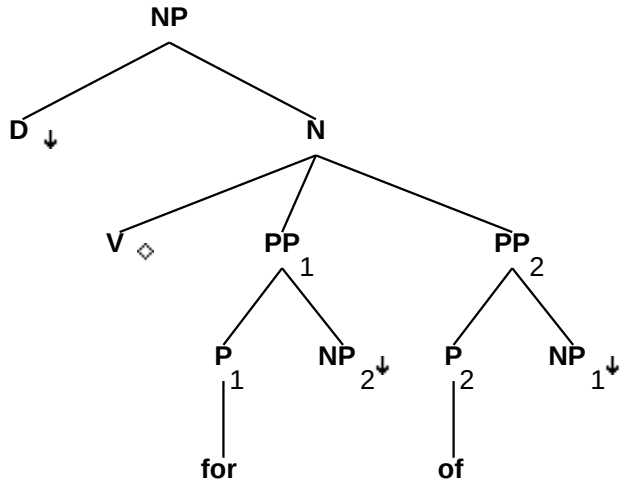
36.3 features

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



VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-case> = V.t:<assign-case>
```

```
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
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>
```

37 Tree "alphaDnx0Vnx2nx1"

37.1 graphe



37.2 comments

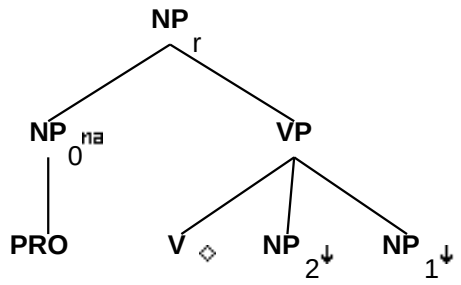
```
Ditransitive determiner gerund tree

Ex:  The winning for her sister of a stuffed animal
```

37.3 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>
P_2.b:<assign-case> = acc
PP_2.b:<assign-case> = P_2.t:<assign-case>
PP_2.b:<assign-case> = NP_2.t:<case>
PP_1.b:<wh> = NP_1:<wh>
PP_2.b:<wh> = NP_2:<wh>

38 Tree "alphaGnx0Vnx2nx1-PRO"

38.1 graphe



38.2 comments

Ditransitive gerund tree w/ PRO subject:

Ex: [PRO asking Mary the question] was a difficult thing to do.

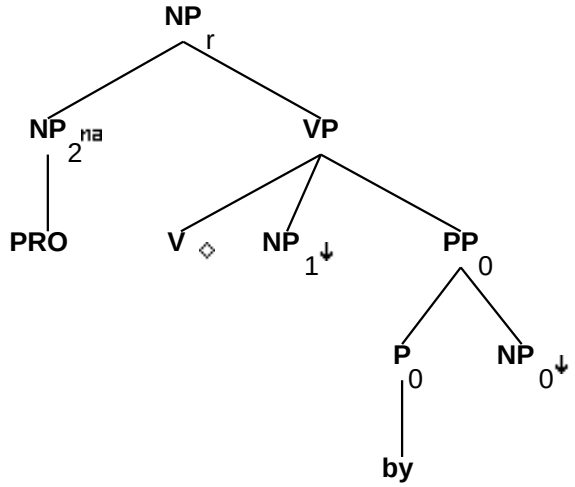
38.3 features

NP_0:<wh> = NP_r.b:<wh>
NP_0.t:<case> = none
NP_0.t:<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
NP_2:<case> = acc
NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -

39 Tree "alphaGnx2Vnx1bynx0-PRO"

39.1 graphe



39.2 comments

Ditransitive gerund passive tree with the 'by' phrase and PRO subject:
'John does not like [PRO being asked questions by the police]'

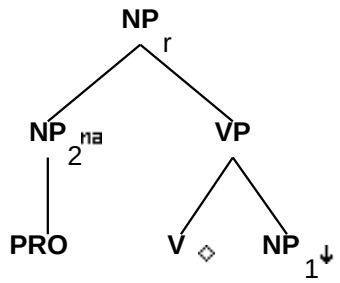
39.3 features

NP_1:<case> = acc
NP_2:<wh> = NP_r.b:<wh>
NP_2.t:<case> = none
NP_2.t:<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_r.b:<gerund> = +
VP.t:<mode> = ger

VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
PP_0.b:<wh> = NP_0:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
V.t:<passive> = +
V.t:<mode> = ppart

40 Tree "alphaGnx2Vnx1-PRO"

40.1 graphe



40.2 comments

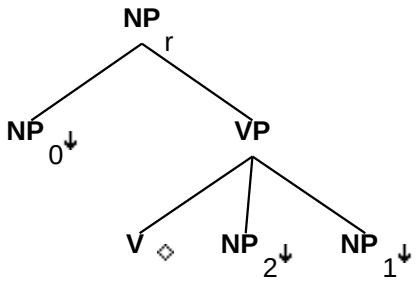
Ditransitive gerund passive tree without the 'by' phrase, w/ PRO subject: 'John does not like [PRO being asked questions]'

40.3 features

NP_1:<case> = acc
NP_2:<wh> = NP_r.b:<wh>
NP_2:<case> = none
NP_2.t:<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_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<mode> = ppart
V.t:<passive> = +

41 Tree "alphaGnx0Vnx2nx1"

41.1 graphe



41.2 comments

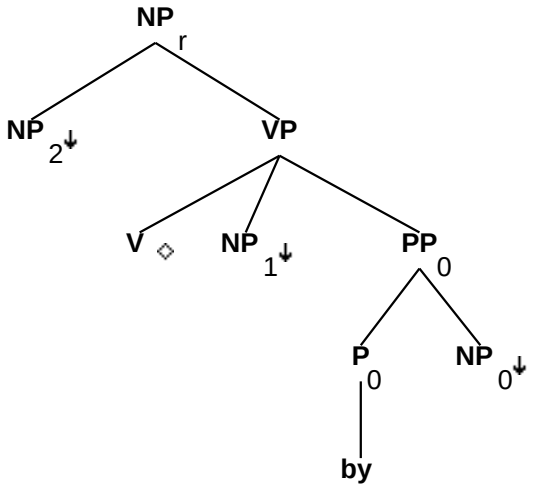
```
Ditransitive gerund tree

Ex:
His asking Mary the question was a difficult thing to do.

Him asking Mary tge question was a difficult thing to do.
```

41.3 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
NP_2:<case> = acc



NP_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<passive> = -
NP_0:<case> = acc/gen
```

42 Tree "alphaGnx2Vnx1bynx0"

42.1 graphe



42.2 comments

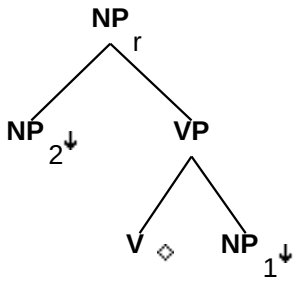
```
Ditransitive gerund passive tree with the 'by' phrase:
'John does not like 'Mary('s) being asked questions by the police''
```

42.3 features

```
NP_1:<case> = acc
NP_2:<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_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
PP_0.b:<wh> = NP_0:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
NP_0:<case> = PP_0.b:<assign-case>
V.t:<passive> = +
V.t:<mode> = ppart
NP_2:<case> = acc/gen
```

43 Tree "alphaGnx2Vnx1"

43.1 graphe



43.2 comments

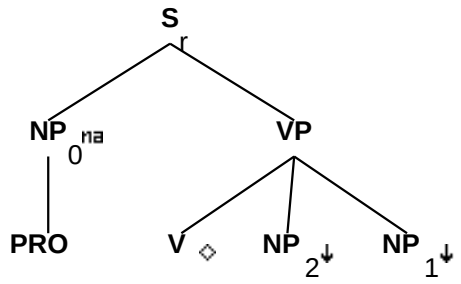
```
Ditransitive gerund passive tree without the 'by' phrase:
'John does not like 'Mary('s) being asked questions''
```

43.3 features

```
NP_1:<case> = acc
NP_2:<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_r.b:<gerund> = +
VP.t:<mode> = ger
VP.b:<compar> = -
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
V.t:<mode> = ppart
V.t:<passive> = +
NP_2:<case> = acc/gen
```

44 Tree "alphanx0Vnx2nx1-PRO"

44.1 graphe



44.2 comments

```
Ditransitive w/ PRO subject:

Bill doesn't want [PRO to give them presents].
While [PRO giving them presents] Bill realized he didn't like them.
```

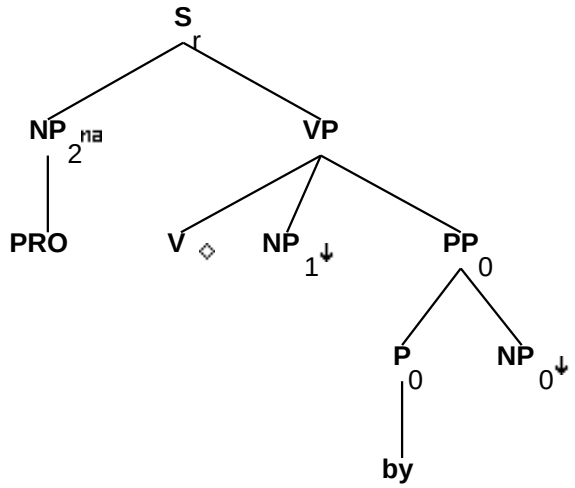
44.3 features

```
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:<wh> = NP_0:<wh>
S_r.b:<assign-case> = NP_0.t:<case>
NP_0:<agr> = S_r.b:<agr>
NP_0:<wh> = -
NP_0.t:<case> = none
NP_1:<case> = acc
NP_2:<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:<mode> = V.t:<mode>
VP.b:<tense> = V.t:<tense>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
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 "alphanx2Vnx1bynx0-PRO"

45.1 graphe



45.2 comments

Ditransitive passive tree w/ PRO subject:

Mary wanted [PRO to be asked a questionby John].
While [PRO being asked a question by John] Mary went deaf.

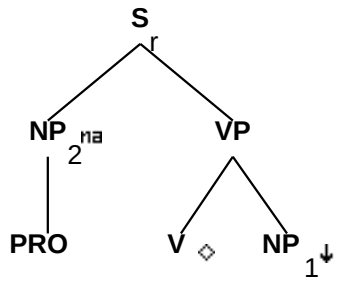
45.3 features

S_r.b:<inv> = -
S_r.b:<extracted> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
S_r.b:<assign-case> = NP_2.t:<case>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_2.t:<case> = none
NP_1:<case> = acc

```
S_r.b:<agr> = VP.t:<agr>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-comp> = V.t:<assign-comp>
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>
P_0.b:<assign-case> = acc
S_r.b:<control> = NP_1.t:<control>
PP_0.b:<wh> = NP_0:<wh>
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 "alphanx2Vnx1-PRO"

46.1 graphe



46.2 comments

```
Ditransitive passive w/out by-phrase, w/ PRO subject:

Mary wanted [PRO to be asked a question].
While [PRO being asked a question] Mary went deaf.
```

46.3 features

```
S_r.b:<extracted> = -
S_r.b:<inv> = -
S_r.b:<assign-comp> = VP.t:<assign-comp>
VP.b:<mode> = V.t:<mode>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<comp> = nil
```

```
S_r.b:<tense> = VP.t:<tense>
S_r.b:<wh> = NP_2:<wh>
S_r.b:<assign-case> = NP_2.t:<case>
NP_2:<agr> = S_r.b:<agr>
NP_2:<case> = S_r.b:<assign-case>
NP_2:<wh> = -
NP_2.t:<case> = none
NP_1:<case> = acc
S_r.b:<agr> = VP.t:<agr>
VP.b:<agr> = V.t:<agr>
VP.b:<tense> = V.t:<tense>
VP.b:<assign-comp> = V.t:<assign-comp>
VP.b:<mainv> = V.t:<mainv>
VP.b:<compar> = -
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
```