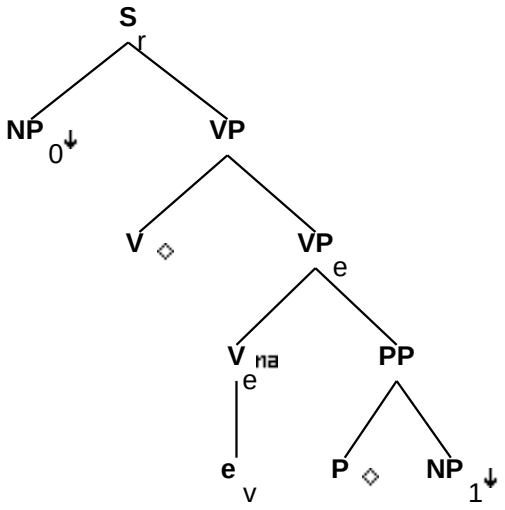
Family "Tnx0VPnx1"

March 5, 2008

1 Tree "alphanx0VPnx1"

1.1 graphe



1.2 comments

Basic declarative tree for verbs which take particular prepositional complements.

John depends on Mary.
John thought of a new idea.

The VP_e node allows left adjunction between the verb and the preposition:

John depends often on Mary.

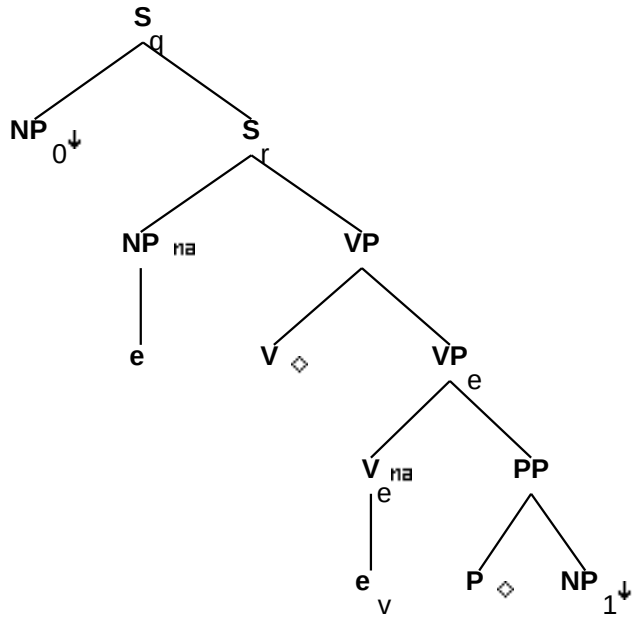
1.3 features

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> = -
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> = VP_e.t:<compar>
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

S_r.b:<control> = NP_0.t:<control>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

2 Tree "alphaW0nx0VPnx1"

2.1 graphe



2.2 comments

Wh on the subject

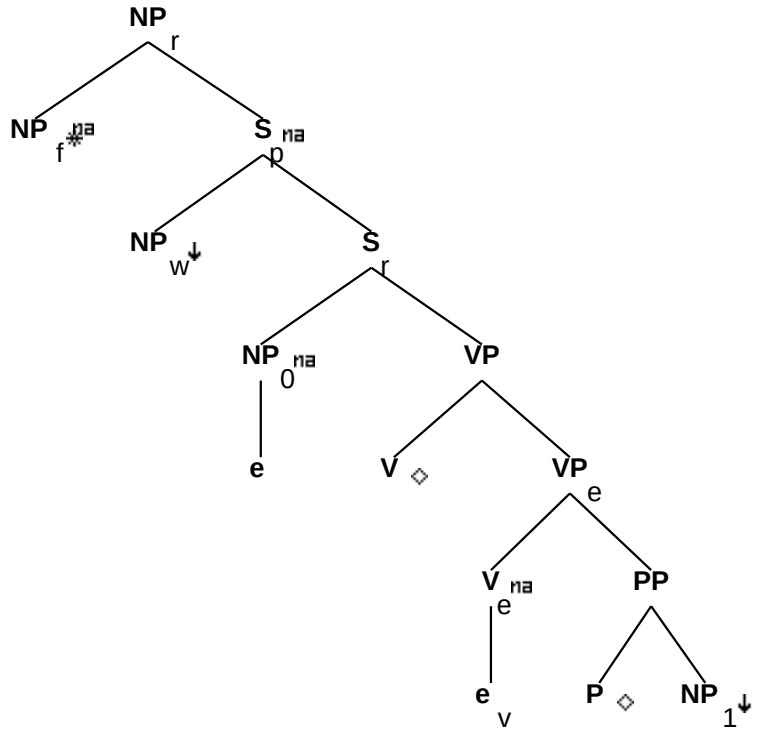
2.3 features

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

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<wh> = NP_0.t:<wh>
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>



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:<trace> = NP_0.t:<trace>
NP:<agr> = NP_0.t:<agr>
NP:<case> = NP_0.t:<case>
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>
S_r.b:<agr> = NP.t:<agr>
S_r.b:<assign-case> = NP.t:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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

P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
```

3 Tree "betaN0nx0VPnx1"

3.1 graphe



3.2 comments

```
Subject relative clause:
'...the man who depends on Mary...'
```

3.3 features

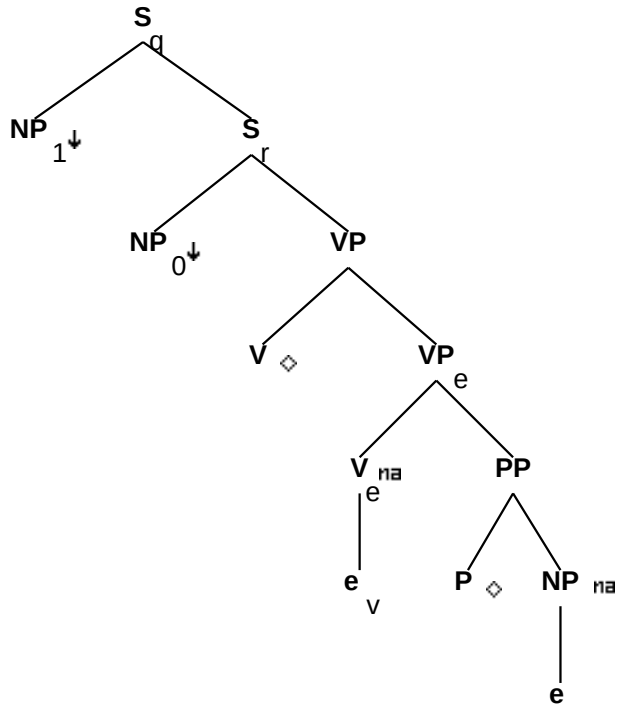
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

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

```
S_r.b:<agr> = NP_0.t:<agr>
S_r.b:<assign-case> = NP_0.t:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

4 Tree ”alphaW1nx0VPnx1”

4.1 graphe



4.2 comments

```
Wh & topicalization on NP1:
   'Who does John depend on?'
   'Mary John depends on. (Emma he doesn't)'
```

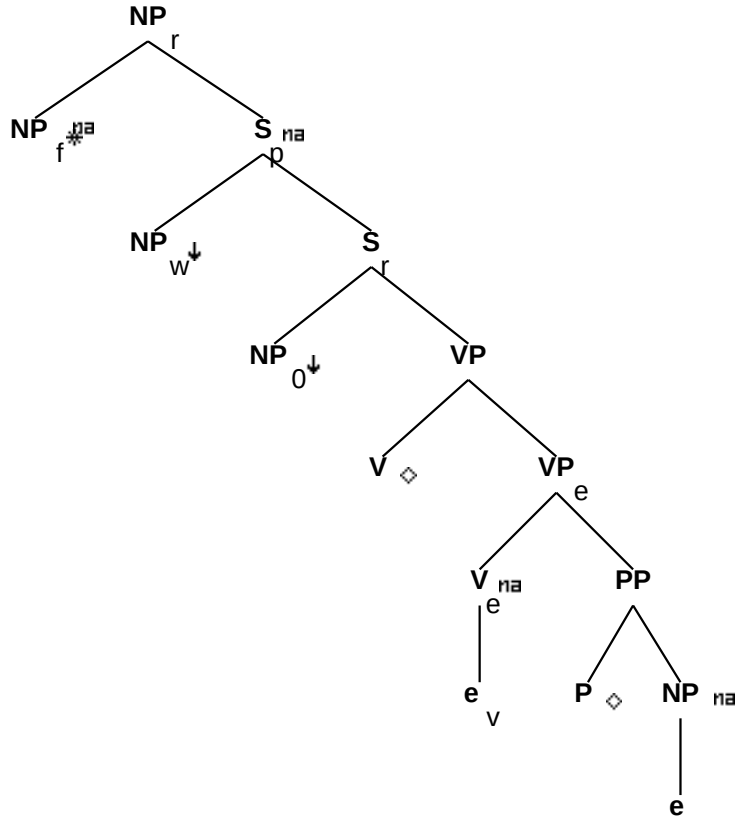
4.3 features

```
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:<comp> = nil
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:<tense> = VP.t:<tense>
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<assign-case>
NP_1:<wh> = S_q.b:<wh>
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:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP:<wh>
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 "betaN1nx0VPnx1"

5.1 graphe



5.2 comments

```
Object relative:
    '...the woman that John depends on...'
```

5.3 features

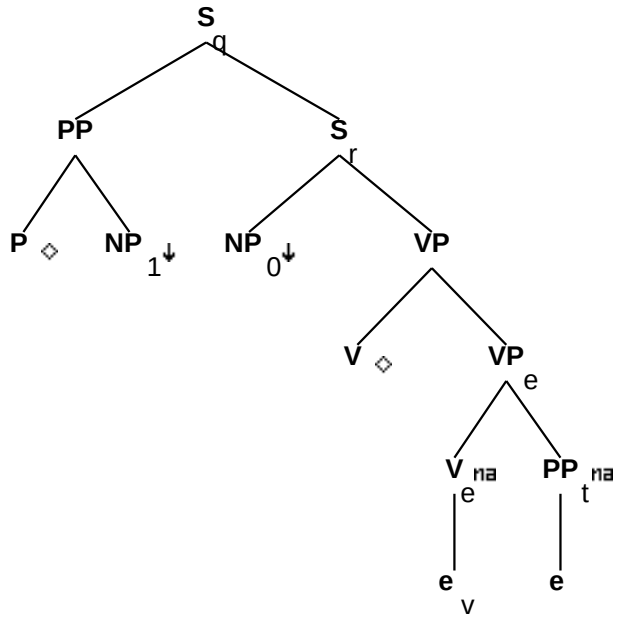
```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_r.t:<mode> = ind/inf
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
```

```
NP_0:<case> = S_r.b:<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:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP_1:<wh>
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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 "alphapW1nx0VPnx1"

6.1 graphe



6.2 comments

```
Fronted PP-complement:
'On these they feel they can rely.' (Brown corpus)
```

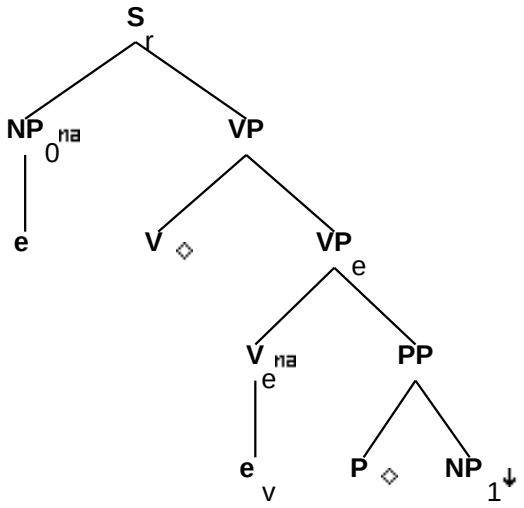
6.3 features

```
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:<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>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<assign-case> = VP.t:<assign-case>
PP_1:<trace> = PP.t:<trace>
```

```
S_q.b:<wh> = PP_1:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

S_r.t:<conj> = nil
S_r.b:<control> = NP_0.t:<control>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

7 Tree "alphaInx0VPnx1"

7.1 graphe



7.2 comments

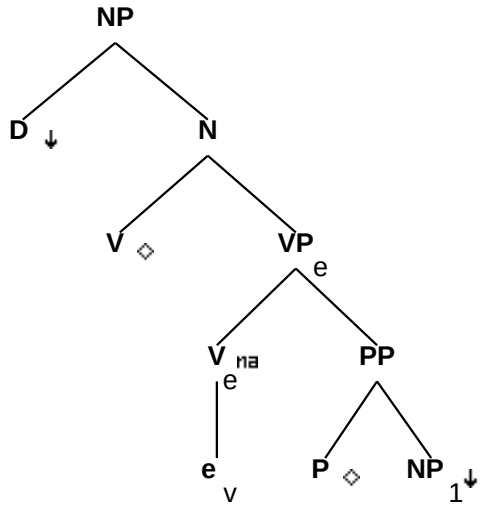
```
Imperative:
    'Think of an answer!'
```

7.3 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
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<assign-comp> = none

V.b:<mode> = base
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

8 Tree "alphaDnx0VPnx1"

8.1 graphe



8.2 comments

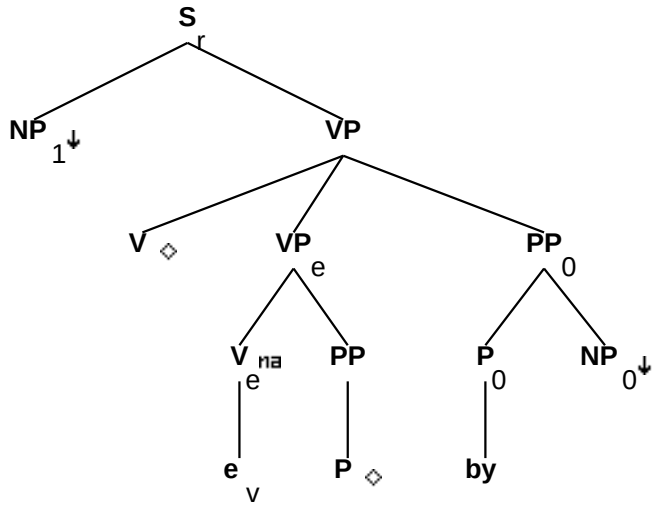
```
PP-complement: determiner gerund
   'His depending entirely on Mary bothered me greatly.'
```

8.3 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.t:<mode> = ger
NP.b:<case> = nom/acc
NP.b:<agr num> = sing
NP.b:<agr pers> = 3
NP.b:<agr 3rdsing> = +
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
```

9 Tree "alphanx1VPbynx0"

9.1 graphe



9.2 comments

```
Passive with by-phrase:
    'The idea was thought of by John.'
```

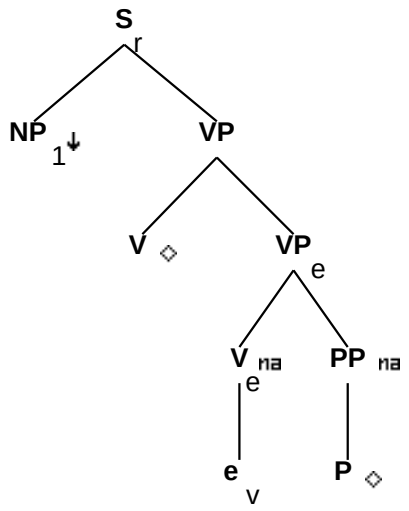
9.3 features

```
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_1:<wh>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
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.b:<control> = NP.t:<control>
PP_0.b:<wh> = NP_0:<wh>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

10 Tree "alphanx1VP"

10.1 graphe



10.2 comments

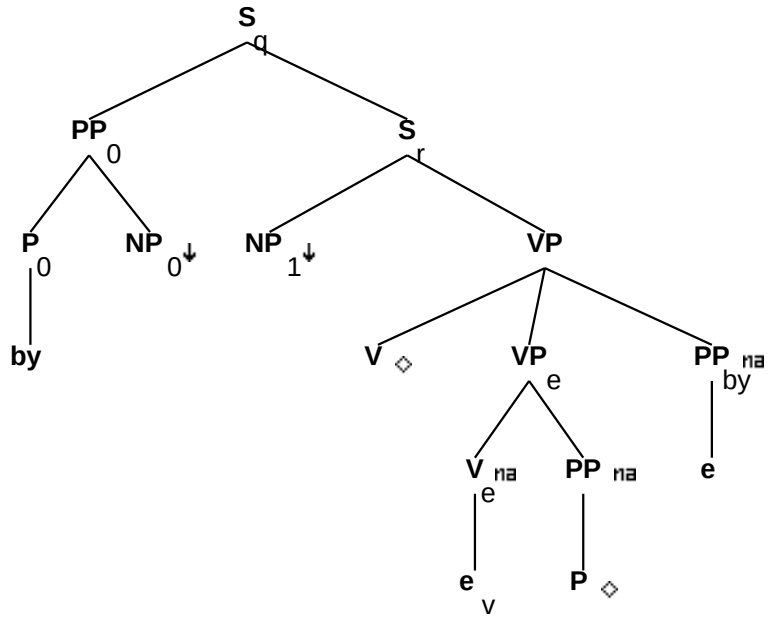
```
Passive:
   'The idea was thought of.'
```

10.3 features

```
S_r.b:<inv> = -
S_r.b:<comp> = nil
S_r.b:<extracted> = -
S_r.b:<wh> = NP_1:<wh>
S_r.b:<agr> = NP_1:<agr>
S_r.b:<assign-case> = NP_1:<case>
S_r.b:<control> = NP_1.t:<control>
S_r.b:<agr> = VP.t:<agr>
S_r.b:<mode> = VP.t:<mode>
S_r.b:<mainv> = VP.t:<mainv>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<perfect> = VP.t:<perfect>
S_r.b:<passive> = VP.t:<passive>
S_r.b:<progressive> = VP.t:<progressive>
S_r.b:<assign-case> = VP.t:<assign-case>
S_r.b:<assign-comp> = VP.t:<assign-comp>


NP_1:<wh> = -

VP.b:<compar> = -

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

VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none


V.t:<passive> = +
V.t:<mode> = ppart
V.t:<punct struct> = nil
```

11 Tree "alphapW0nx1VPbynx0"

11.1 graphe



11.2 comments

```
Wh question on NP0 in passive constructions, by-phrase extracted:
    'By whom was the idea thought of?'
```

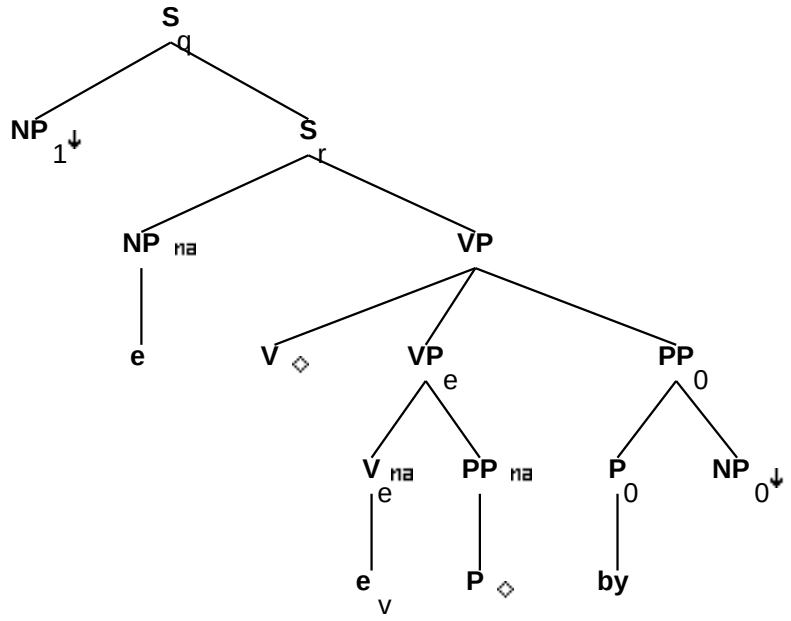
11.3 features

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

S_q.b:<inv> = S_r.t:<inv>
S_q.b:<inv> = S_q.b:<invlink>
P_0.b:<assign-case> = acc
S_r.t:<comp> = nil
S_r.b:<assign-comp> = VP.t:<assign-comp>
PP_0.b:<assign-case> = P_0.t:<assign-case>



NP_0:<case> = PP_0.b:<assign-case>
PP_0.b:<wh> = NP_0:<wh>
S_q.b:<wh> = PP_0.t:<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.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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
PP.t:<trace> = PP_0.t:<trace>
S_r.t:<conj> = nil
S_r.b:<control> = NP.t:<control>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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 "alphaW1nx1VPbynx0"

12.1 graphe



12.2 comments

```
Wh question on NP1 in passive constructions
    'What was thought of by Bob?'
```

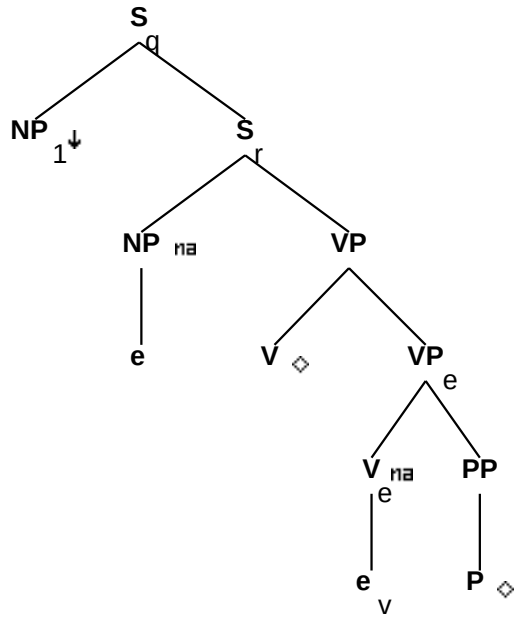
12.3 features

```
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_q.b:<wh> = NP_1:<wh>
S_r.b:<inv> = -
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
NP_1:<agr> = NP.t:<agr>
NP_1:<case> = NP.t:<case>
NP.t:<wh> = +
NP_1:<trace> = NP.t:<trace>
NP_1:<wh> = NP.t:<wh>
```

```
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.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>
V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
```

13 Tree "alphaW1nx1VP"

13.1 graphe



13.2 comments

```
Wh question on NP1 in passive constructions
    'What was thought of?'
```

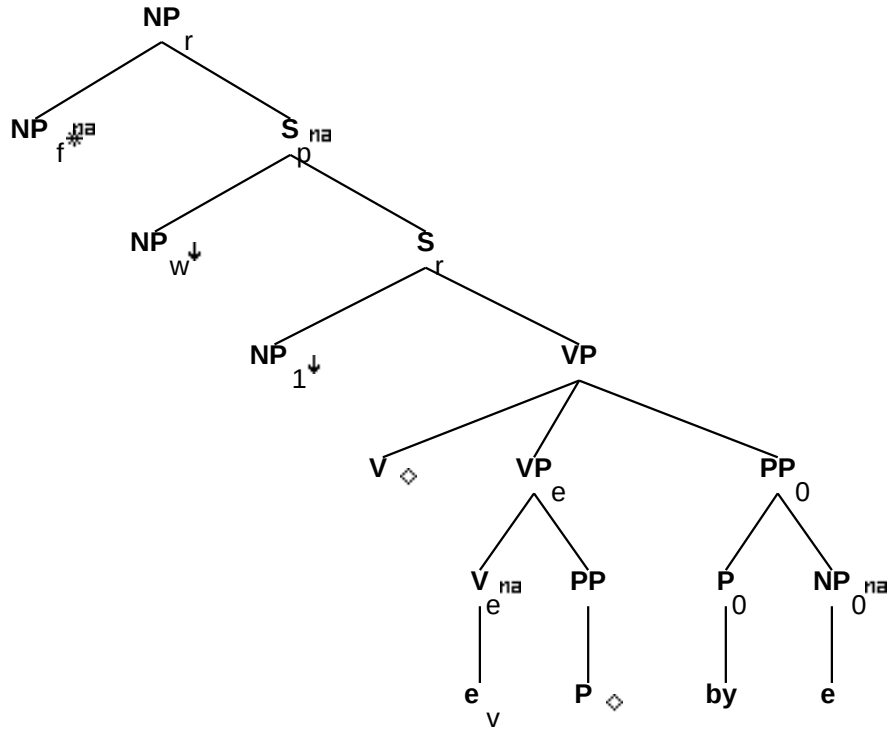
13.3 features

```
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_q.b:<wh> = NP_1:<wh>
S_r.b:<inv> = -
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
NP_1:<agr> = NP.t:<agr>
NP_1:<case> = NP.t:<case>
NP_1:<wh> = NP.t:<wh>
NP.t:<wh> = +
NP_1:<trace> = NP.t:<trace>
```

```
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.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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
VP.b:<passive> = V.t:<passive>
S_r.t:<conj> = nil
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
```

14 Tree "betaN0nx1VPbynx0"

14.1 graphe



14.2 comments

```
That relative clause, extraction of NP0 from by-phrase:
(I know) the person that the idea was thought of by.'
```

14.3 features

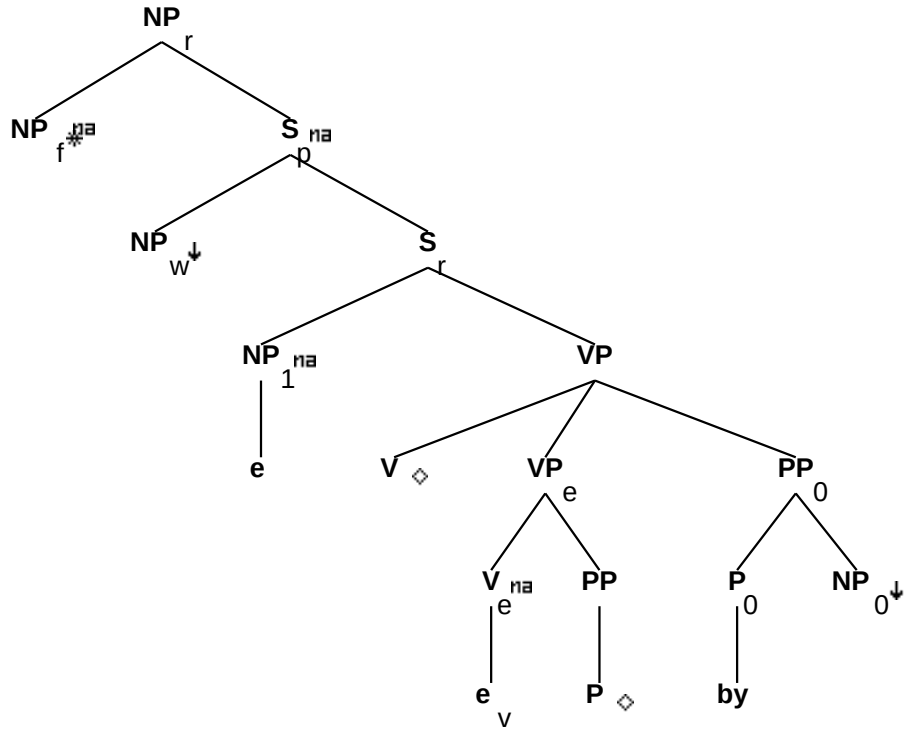
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

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

```
S_r.b:<assign-case> = NP.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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
S_r.t:<conj> = nil

S_r.b:<control> = NP.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>
```

15 Tree "betaN1nx1VPbynx0"

15.1 graphe



15.2 comments

```
That relative clause, extraction from NP1:
   '(I heard) the idea that was thought of by John.'
```

15.3 features

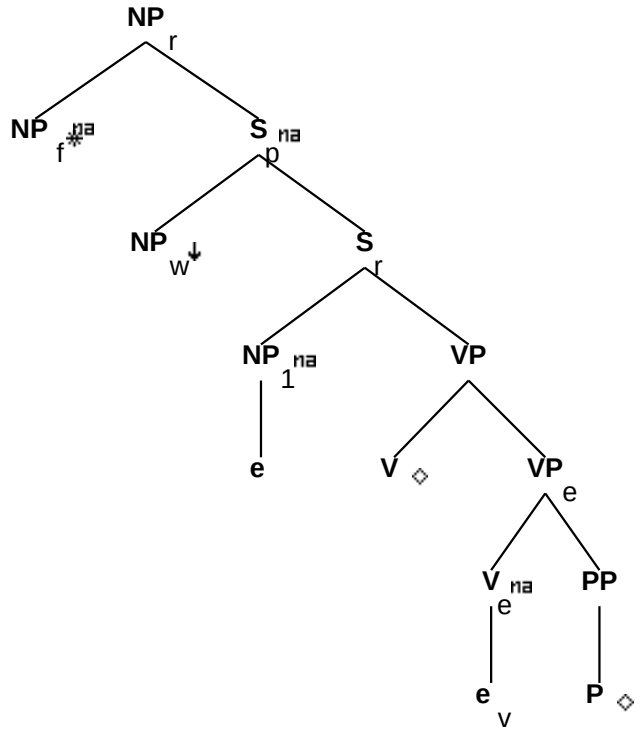
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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.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:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
S_r.t:<conj> = nil

NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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 "betaN1nx1VP"

16.1 graphe



16.2 comments

```
That relative clause, extraction from NP1:
    '(I saw) the woman that was depended on.'
```

16.3 features

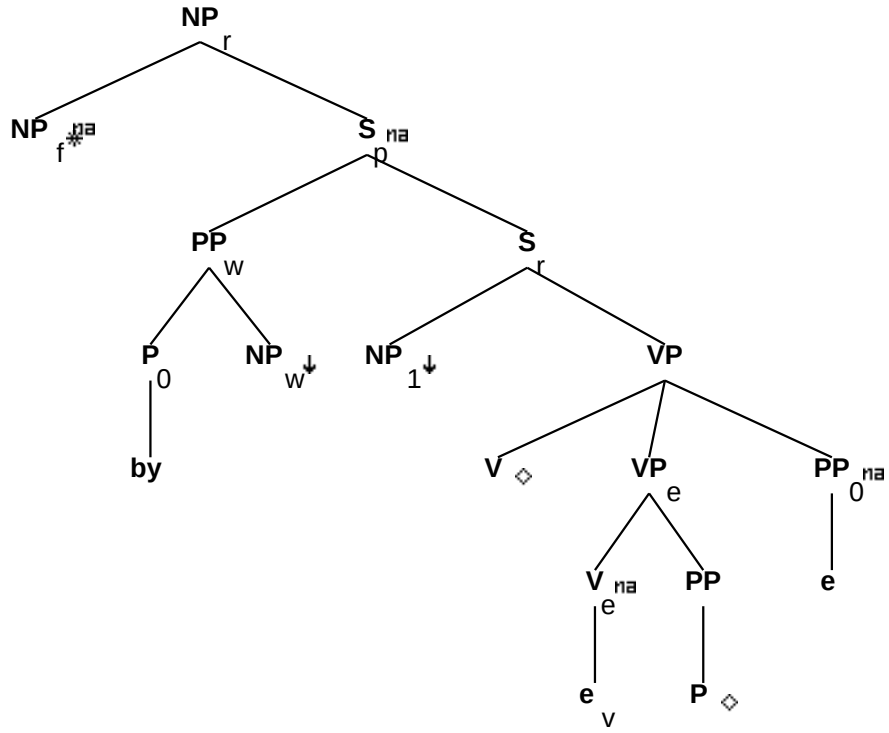
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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.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:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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> = -
S_r.t:<conj> = nil

NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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 "betaNbynx0nx1VPbynx0"

17.1 graphe



17.2 comments

```
That relative clause, extraction of NP0 from by-phrase:
   '(I know) the person by whom the idea was thought of.'
```

17.3 features

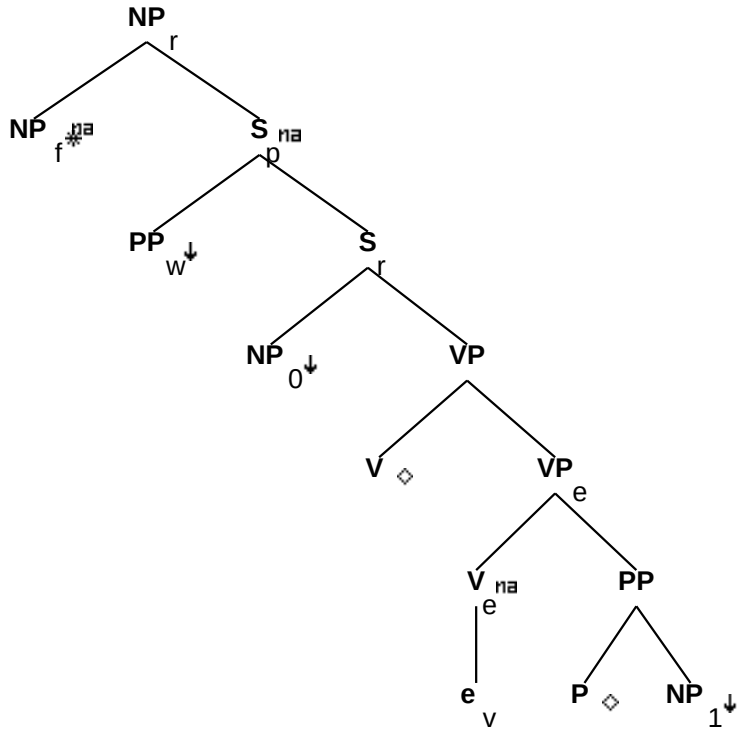
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

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

```
S_r.b:<assign-case> = NP.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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
S_r.t:<conj> = nil

S_r.b:<control> = NP.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>
```

18 Tree "betaNpxnx0VPnx1"

18.1 graphe



18.2 comments

```
Adjunct relative clause:
   'the day on which John thought of the idea (was very joyous).'
```

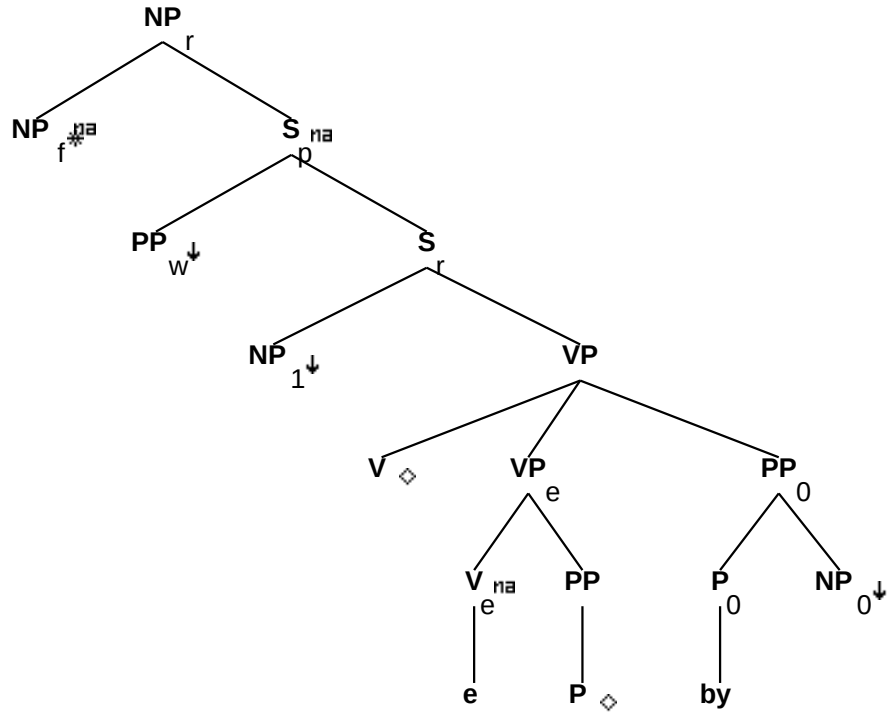
18.3 features

```
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:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

19 Tree "betaNpxnx1VPbynx0"

19.1 graphe



19.2 comments

```
Adjunct relative clause on passive:
   'the day on which the idea was thought of by John'
```

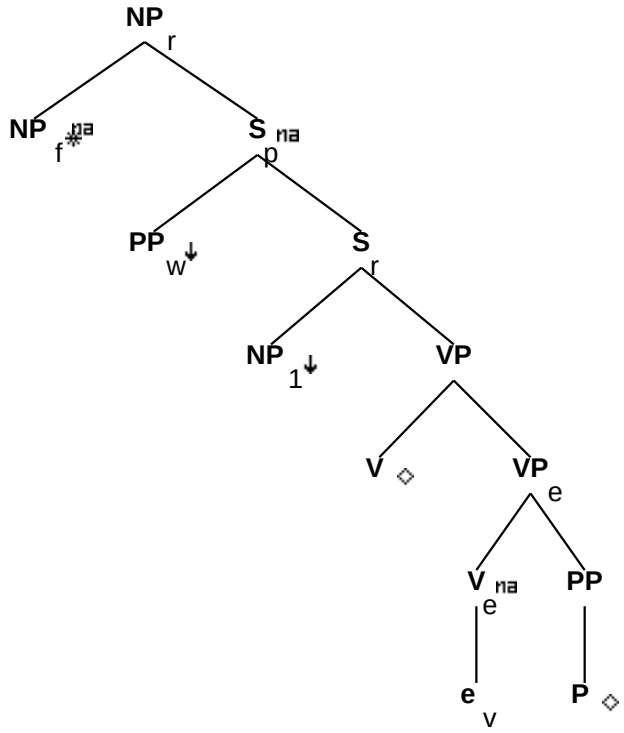
19.3 features

```
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
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.b:<control> = NP.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>
```

20 Tree "betaNpxnx1VP"

20.1 graphe



20.2 comments

```
Adjunct extraction from passive:
   'the day on which the idea was thought of (was very joyous).'
```

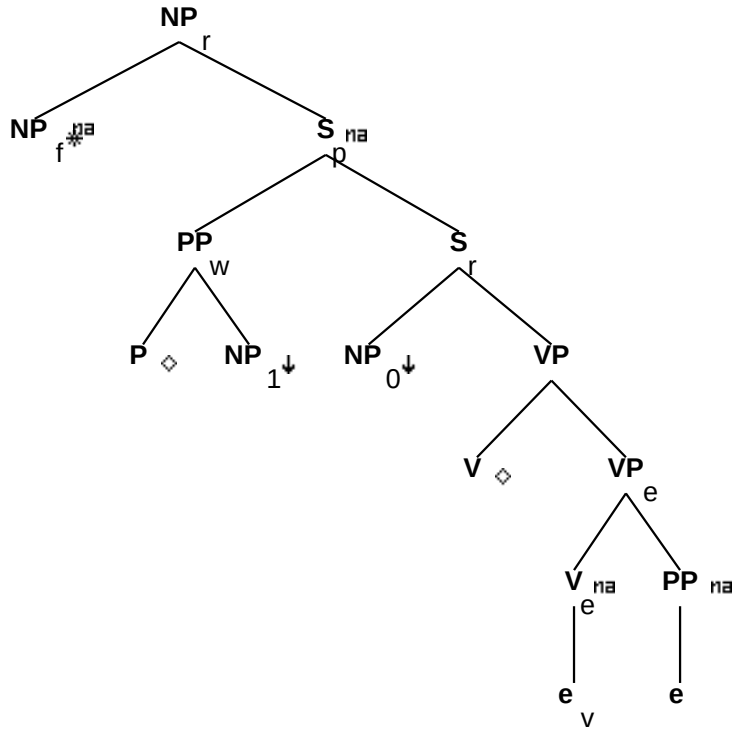
20.3 features

```
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<control> = NP.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>
```

21 Tree "betaNpx1nx0VPnx1"

21.1 graphe



21.2 comments

```
Relative clause on fronted PP:
   'the person on whom John depends (is Mary).'
```

21.3 features

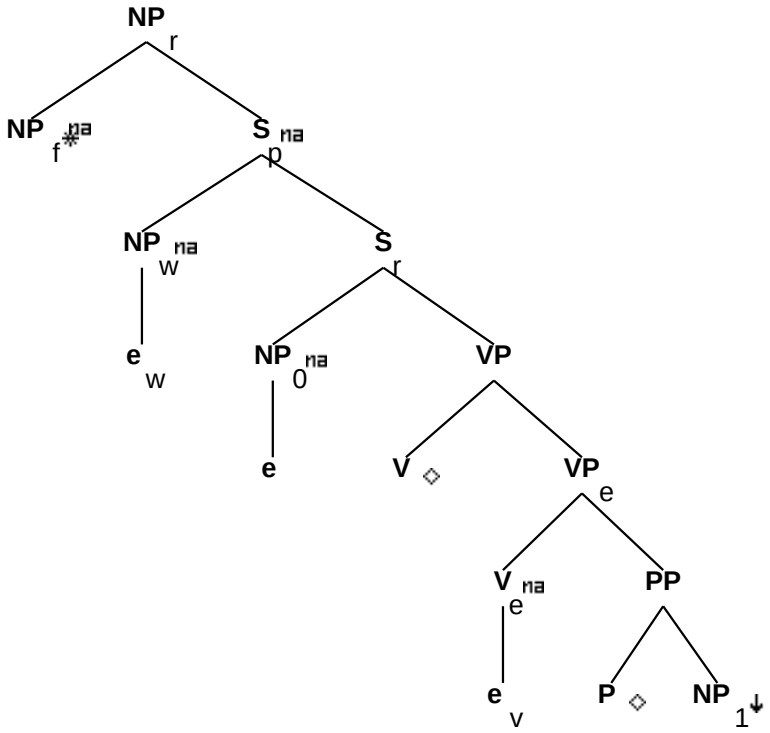
```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_r.t:<mode> = ind/inf
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<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:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

S_r.t:<conj> = nil

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

22 Tree "betaNc0nx0VPnx1"

22.1 graphe



22.2 comments

```
Relative clause on NP0:
  'the man that depends on Mary (slept soundly).'
```

22.3 features

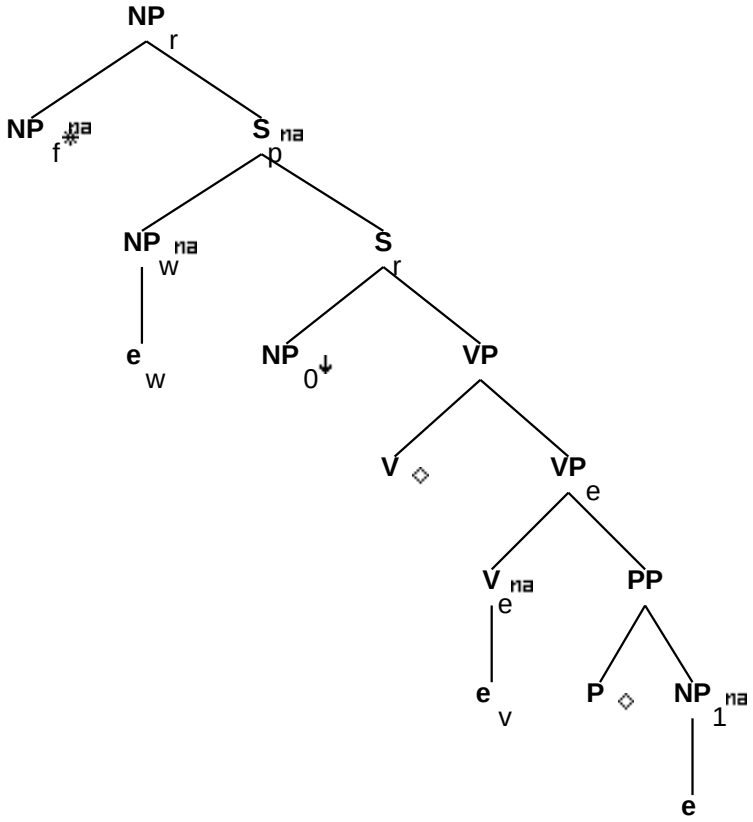
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

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

```
S_r.b:<assign-case> = NP_0.t:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
NP_r.b:<pron> = NP_f.t:<pron>
```

23 Tree "betaNc1nx0VPnx1"

23.1 graphe



23.2 comments

```
Relative clause on NP1:
   'the woman that John depends on (slept soundly).'
```

23.3 features

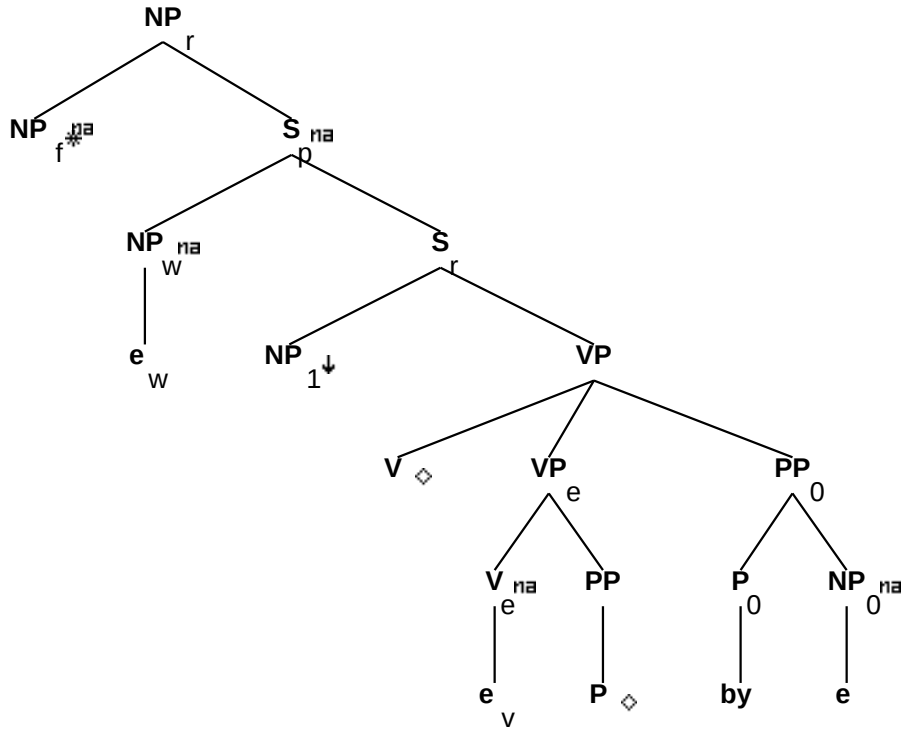
```
S_r.b:<mode> = VP.t:<mode>
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
S_r.b:<tense> = VP.t:<tense>
S_r.t:<inv> = -
S_r.b:<inv> = -
NP_0:<agr> = S_r.b:<agr>
NP_0:<case> = S_r.b:<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:<agr> = VP.t:<agr>
S_r.b:<tense> = VP.t:<tense>
S_r.b:<assign-case> = VP.t:<assign-case>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

PP.b:<assign-case> = P.t:<assign-case>
PP.b:<assign-case> = NP.t:<case>
PP.b:<wh> = NP_1:<wh>
S_r.t:<conj> = nil

S_r.b:<control> = NP_0.t:<control>
NP_w.t:<trace> = NP.b:<trace>
NP_w.t:<case> = NP.b:<case>
NP_w.t:<agr> = NP.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>
```

24 Tree "betaNc0nx1VPbynx0"

24.1 graphe



24.2 comments

```
That relative clause, extraction of NP0 from by-phrase:
   '(I know) the person that the idea was thought of by.'
```

24.3 features

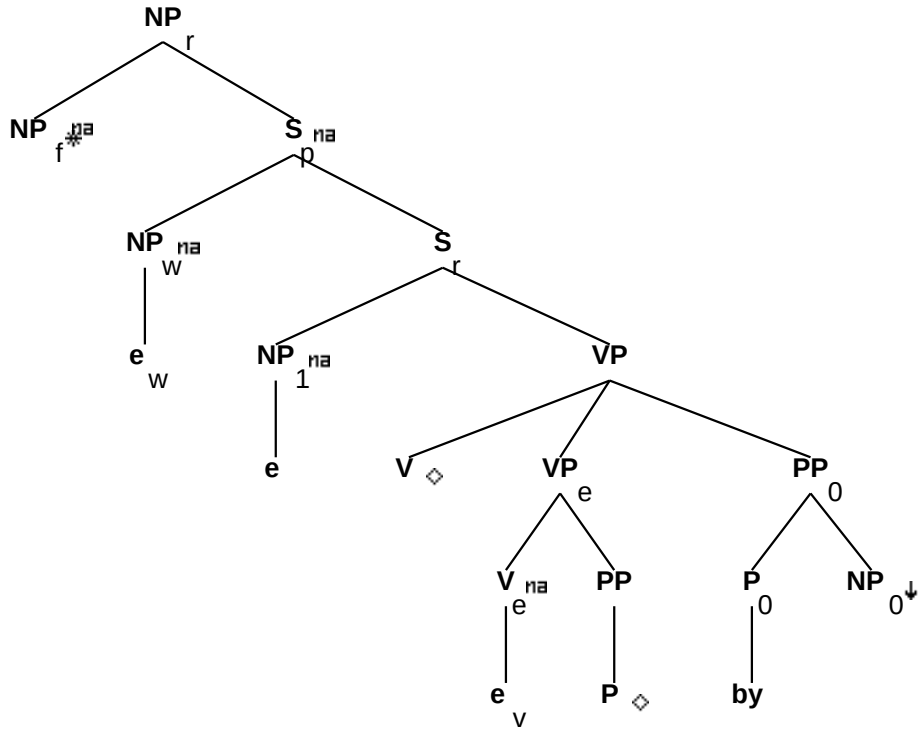
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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.t:<agr>
S_r.b:<assign-case> = NP.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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
S_r.t:<conj> = nil

S_r.b:<control> = NP.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>
```

25 Tree "betaNc1nx1VPbynx0"

25.1 graphe



25.2 comments

```
That relative clause, extraction from NP1:
   '(I saw) the woman that was depended on by John.'
```

25.3 features

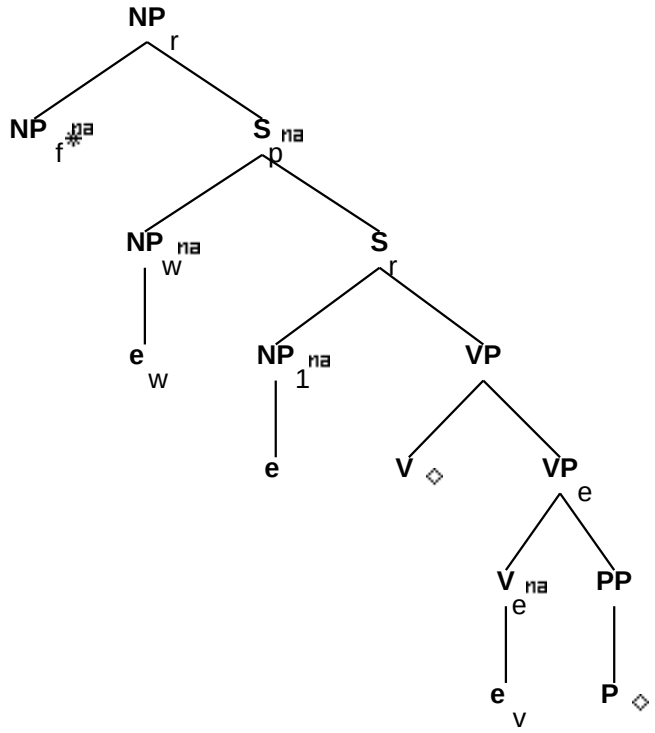
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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.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:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
S_r.t:<conj> = nil

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

26 Tree "betaNc1nx1VP"

26.1 graphe



26.2 comments

```
That relative clause, extraction from NP1:
   '(I saw) the woman that was depended on.'
```

26.3 features

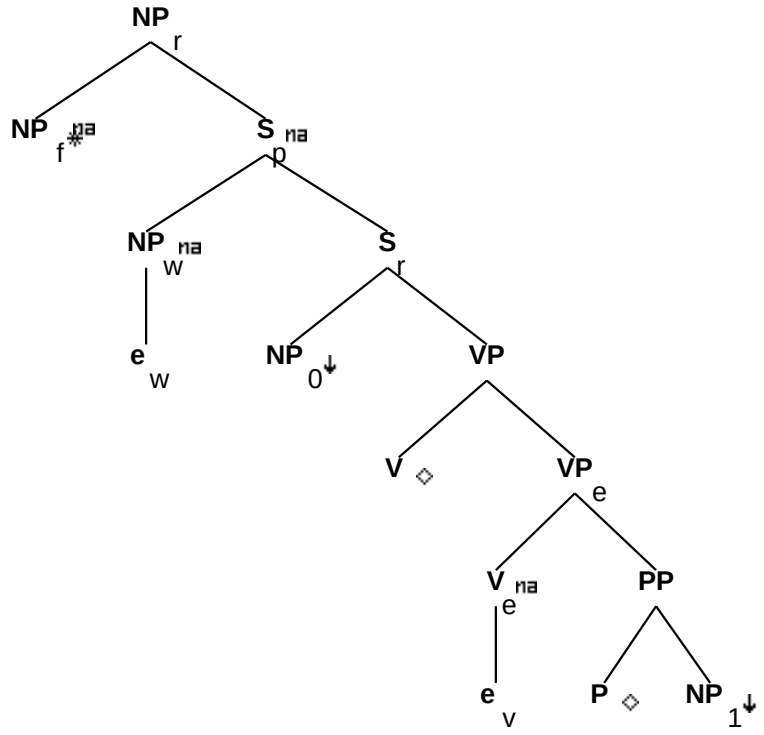
```
S_r.b:<assign-comp> = VP.t:<assign-comp>
```

```
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.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:<mainv> = V.t:<mainv>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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> = -
S_r.t:<conj> = nil

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

27 Tree "betaNcnx0VPnx1"

27.1 graphe



27.2 comments

```
Adjunct relative clause:
   'the day that John thought of the idea (was joyous).'
```

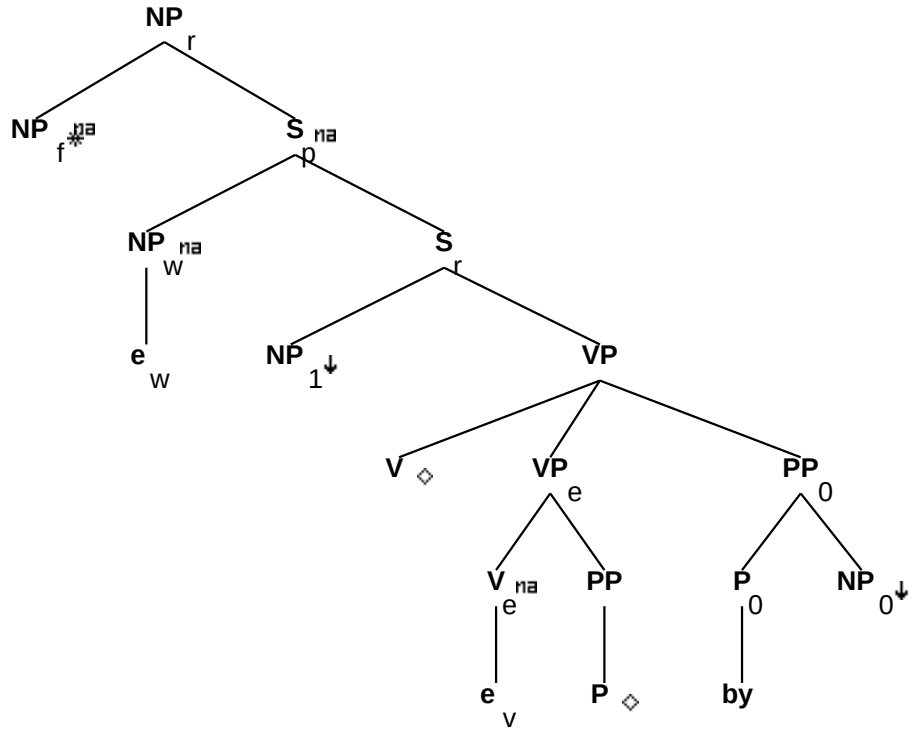
27.3 features

```
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:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

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
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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>
```

28 Tree "betaNcnx1VPbynx0"

28.1 graphe



28.2 comments

```
Adjunct relative clause with passive:
   '(I remember) the day that the idea was thought of by John.'
```

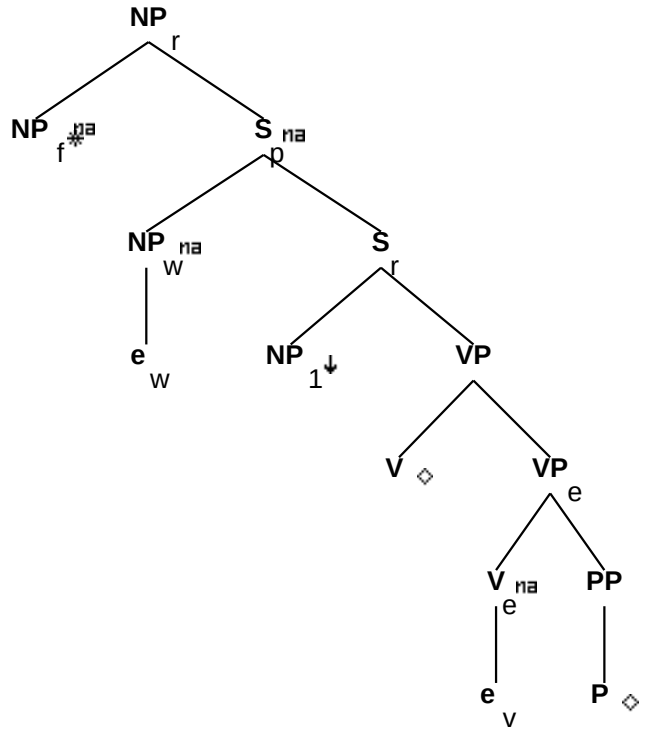
28.3 features

```
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
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.b:<control> = NP.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>
```

29 Tree "betaNcnx1VP"

29.1 graphe



29.2 comments

```
Adjunct relative clause with passive:
   '(I remember) the day that the idea was thought of.'
```

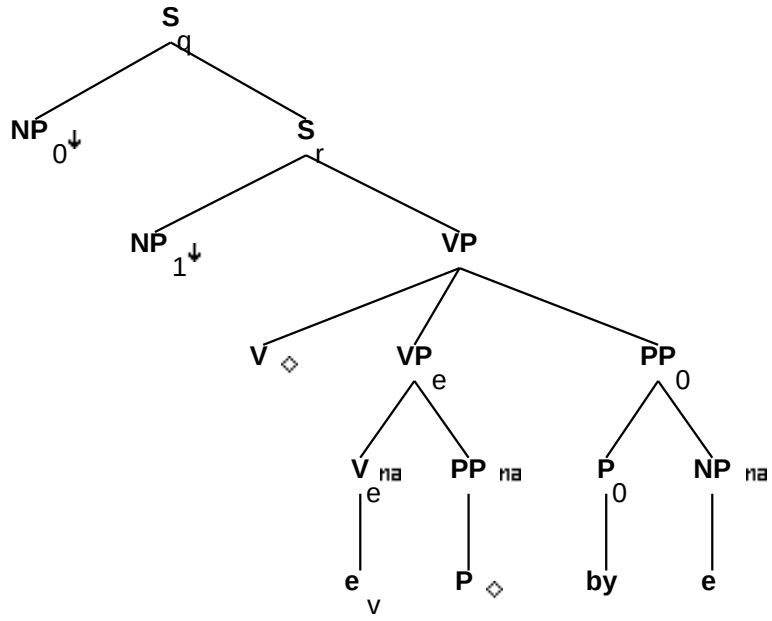
29.3 features

```
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
S_r.b:<control> = NP.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>
```

30 Tree "alphaW0nx1VPbynx0"

30.1 graphe



30.2 comments

```
Wh & topicalization on passivized NP0:
  'Who was the idea thought of by?'
  'John the idea was thought of by.'
```

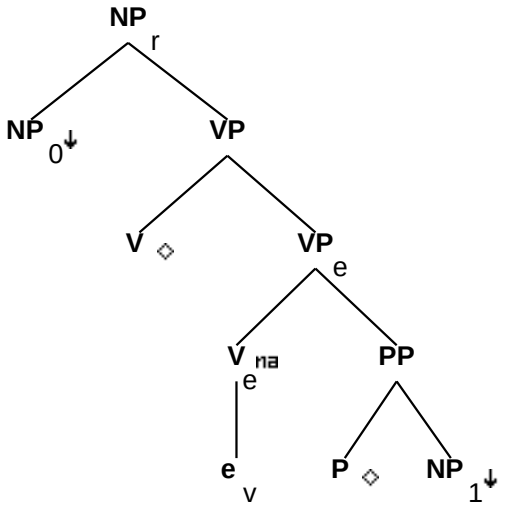
30.3 features

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

```
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
S_r.b:<control> = NP_1:<control>
NP_1:<agr> = S_r.b:<agr>
NP_1:<case> = S_r.b:<assign-case>
V.t:<mode> = ppart
V.t:<passive> = +
V.t:<punct struct> = nil
PP_0.b:<assign-case> = P_0.t:<assign-case>
PP_0.b:<assign-case> = NP.t:<case>
PP_0.b:<wh> = NP.t:<wh>
P_0.b:<assign-case> = acc

S_q.b:<wh> = NP_0:<wh>
S_q.b:<extracted> = +
S_q.b:<inv> = S_q.b:<invlink>
S_q.b:<inv> = S_r.t:<inv>
S_q.b:<mode> = S_r.t:<mode>
S_q.b:<comp> = nil
S_r.t:<comp> = nil
S_r.t:<conj> = nil
V.t:<punct struct> = nil
NP.t:<trace> = NP_0.t:<trace>
NP.t:<agr> = NP_0.t:<agr>
NP.t:<case> = NP_0.t:<case>
NP.t:<wh> = NP_0.t:<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>
```

31 Tree "alphaGnx0VPnx1"

31.1 graphe



31.2 comments

```
Multi Anchor PP-complement - NP gerund

[John('s) depending on Mary] was difficult for her mother.
```

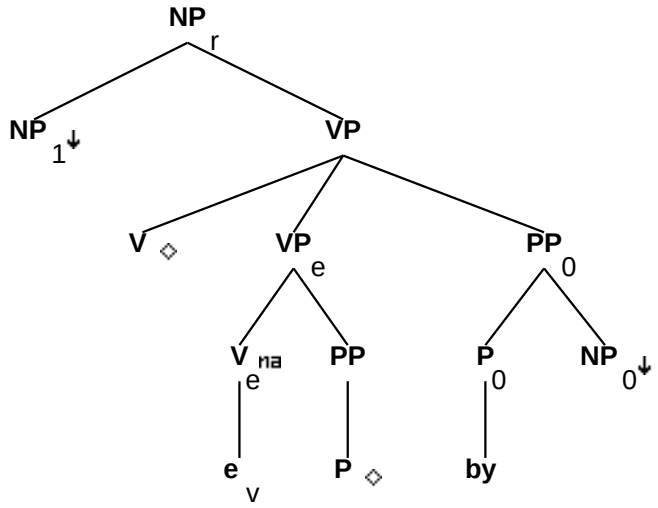
31.3 features

```
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_0:<wh> = NP_r.b:<wh>
VP.t:<mode> = ger

P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<passive> = -
```

NP_0:<case> = acc/gen

32 Tree "alphaGnx1VPbynx0"

32.1 graphe



32.2 comments

Multi Anchor PP-complement - gerund passive with the by-phrase

...the idea('s) being thought of by John...

32.3 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

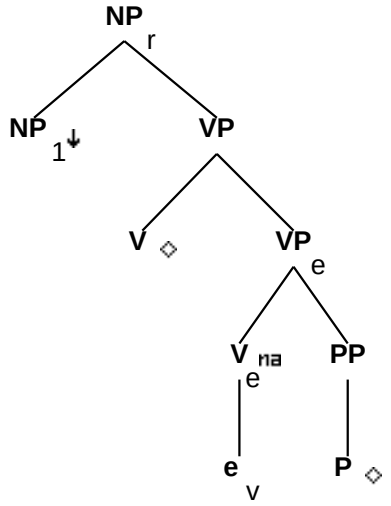
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart

V.t:<passive> = +
PP.b:<wh> = NP_1:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>
NP_0:<case> = PP_0.b:<assign-case>
NP_1:<case> = acc/gen

33 Tree "alphaGnx1VP"

33.1 graphe



33.2 comments

Multi Anchor PP-complement - gerund passive without the by-phrase:

...John('s) being depended on...

33.3 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> = +

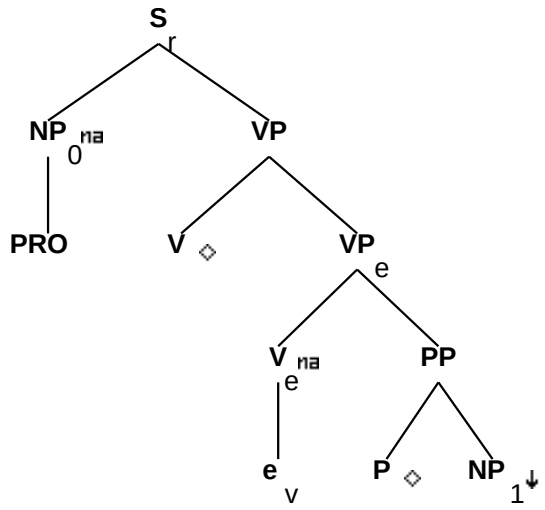
NP_1:<wh> = NP_r.b:<wh>
VP.t:<mode> = ger

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

```
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<wh> = NP_1:<wh>
NP_1:<case> = acc/gen
```

34 Tree "alphanx0VPnx1-PRO"

34.1 graphe



34.2 comments

```
Multi Anchor PP complement - PRO subject


John wants [PRO to think of a new idea].
```

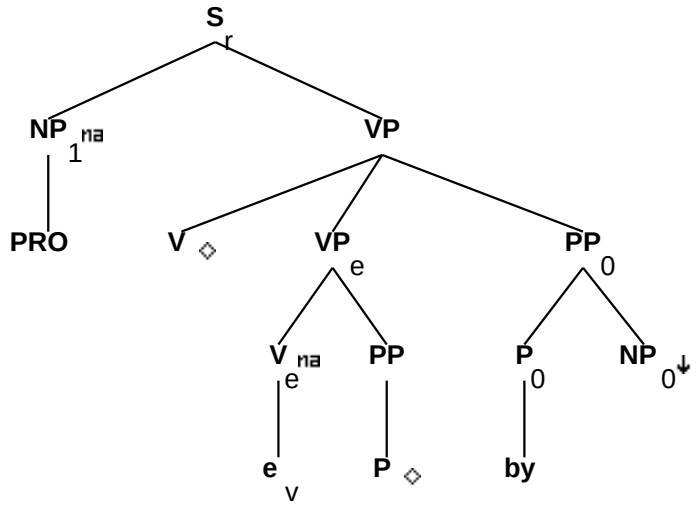
34.3 features

```
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>
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
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> = VP_e.t:<compar>
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

S_r.b:<control> = NP_0.t:<control>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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
```

35 Tree "alphanx1VPbynx0-PRO"

35.1 graphe



35.2 comments

Multi Anchor PP complement - Passive with by-phrase, w/ PRO subject

John wanted [PRO to be depended on by his kids].

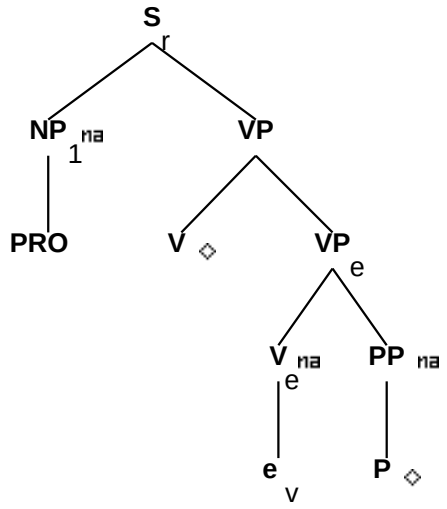
35.3 features

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>
S_r.b:<wh> = NP_1:<wh>
S_r.b:<assign-case> = NP_1.t:<case>
NP_1:<agr> = S_r.b:<agr>
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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
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.b:<control> = NP.t:<control>
PP_0.b:<wh> = NP_0:<wh>
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<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

36 Tree "alphanx1VP-PRO"

36.1 graphe



36.2 comments

```
Multi Anchor PP complement - Passive w/o by-phrase, w/ PRO subject

John wanted [PRO to be depended on].
```

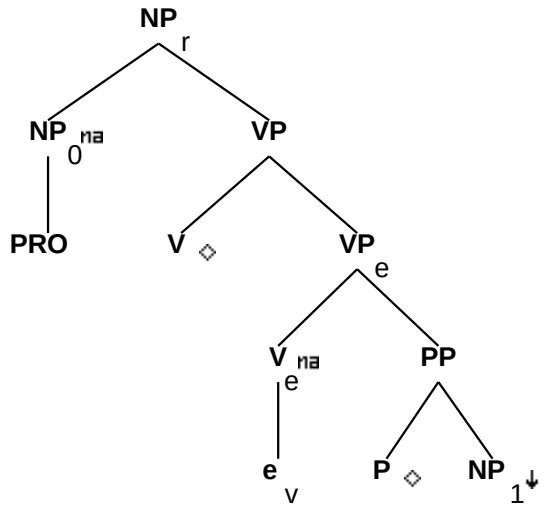
36.3 features

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

```
VP.b:<tense> = V.t:<tense>
VP.b:<passive> = V.t:<passive>
VP.b:<assign-comp> = V.t:<assign-comp>
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<passive> = +
V.t:<mode> = ppart
V.t:<punct struct> = nil
VP.t:<mode> = inf/ger
```

37 Tree "alphaGnx0VPnx1-PRO"

37.1 graphe



37.2 comments

```
Multi Anchor PP-complement - NP gerund w/ PRO subject

[PRO thinking of ideas] makes John feel good.
```

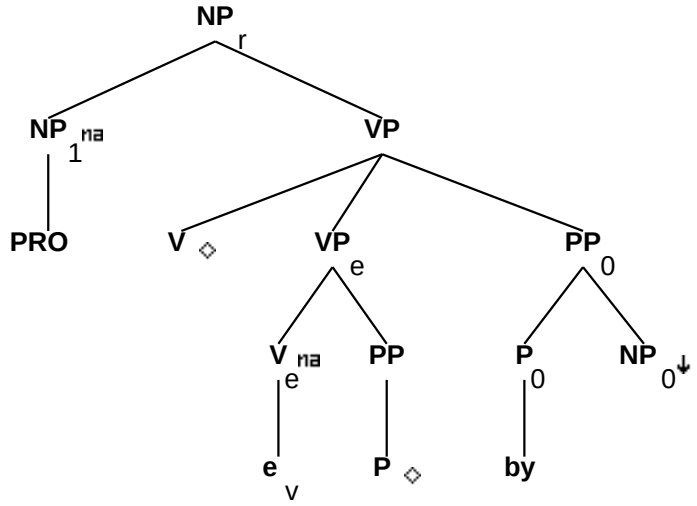
37.3 features

```
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_0:<wh> = NP_r.b:<wh>
NP_0.t:<wh> = -
```

NP_0.t:<case> = none
VP.t:<mode> = ger
P.t:<assign-case> = PP.b:<assign-case>
NP_1:<case> = PP.b:<assign-case>
PP.b:<wh> = NP_1:<wh>
VP.b:<mode> = V.t:<mode>
VP.b:<passive> = V.t:<passive>
VP.b:<compar> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none
V.t:<passive> = -

38 Tree "alphaGnx1VPbynx0-PRO"

38.1 graphe



38.2 comments

Multi Anchor PP-complement - gerund passive with the by-phrase, PRO subject

[PRO being depended on by John] made Mary happy.

38.3 features

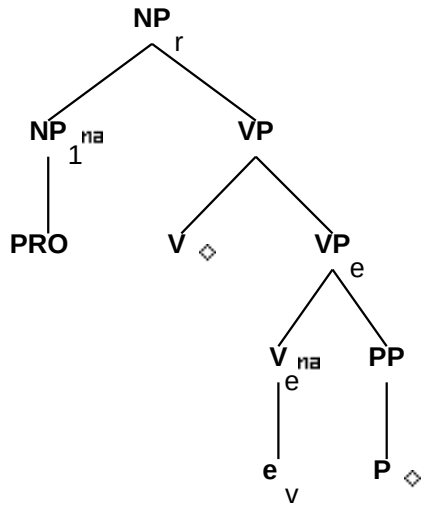
NP_r.b:<case> = nom/acc
NP_r.b:<agr num> = sing
NP_r.b:<agr pers> = 3
NP_r.b:<agr 3rdsing> = +
NP_r.b:<gerund> = +
NP_1:<wh> = NP_r.b:<wh>

NP_1.t:<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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<wh> = NP_1:<wh>
PP_0.b:<assign-case> = P_0.t:<assign-case>
P_0.b:<assign-case> = acc
PP_0.b:<wh> = NP_0:<wh>
NP_0:<case> = PP_0.b:<assign-case>

39 Tree "alphaGnx1VP-PRO"

39.1 graphe



39.2 comments

Multi Anchor PP-complement - gerund passive w/o the by-phrase, w/ PRO subject

[PRO being depended on] made John happy.

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> = -
VP_e.b:<mainv> = -
VP_e.b:<compar> = -
VP_e.b:<mode> = base
VP_e.b:<assign-comp> = none

V.t:<mode> = ppart
V.t:<passive> = +
PP.b:<wh> = NP_1:<wh>
```